KHOA CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

LẬP TRÌNH WEB

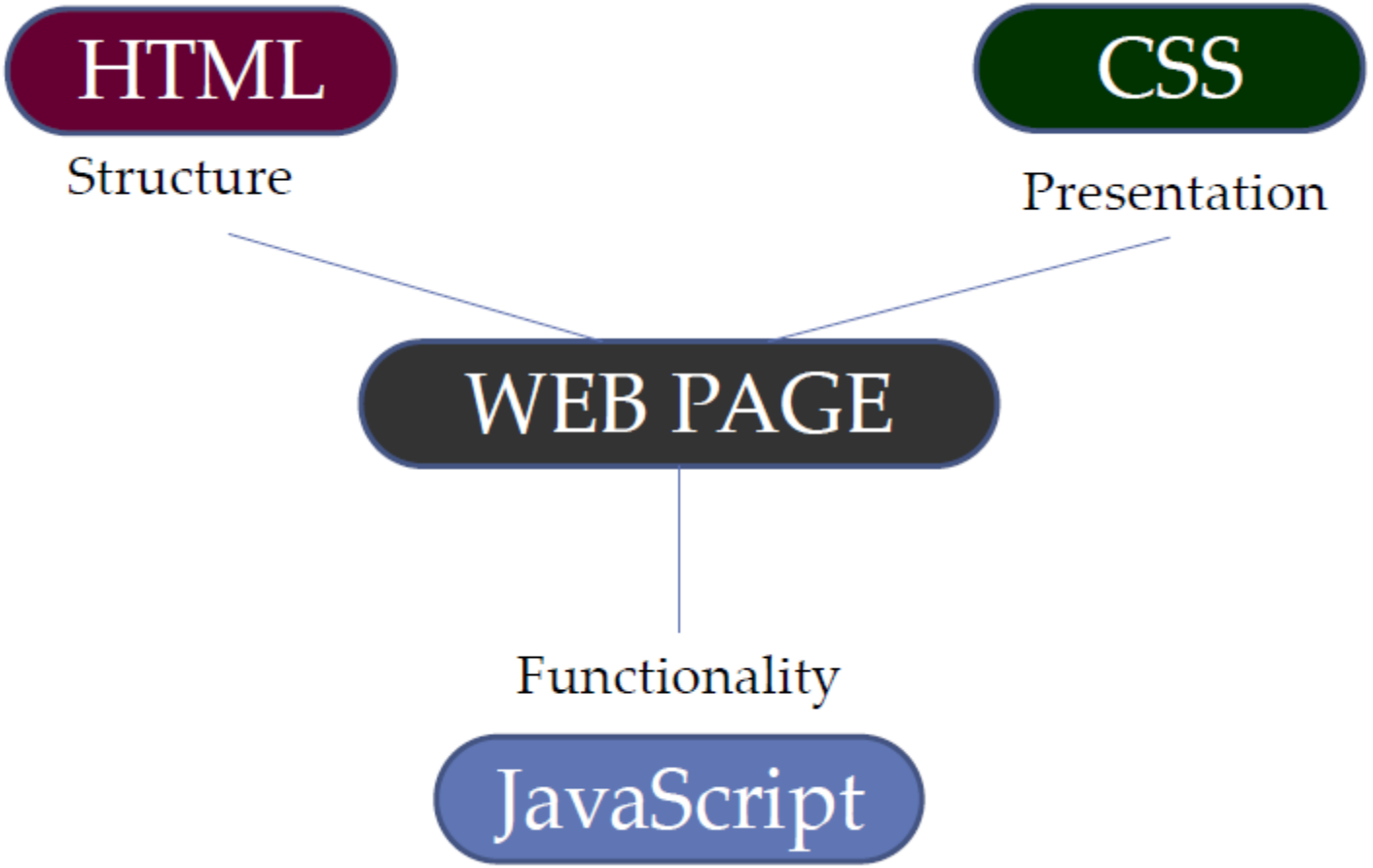
Chương 6: Javascript

Biên soạn : ThS. HÀ DUY AN
Cập nhật : ThS. NGUYỄN CAO HỒNG NGỌC

Nội dung

- Giới thiệu Javascript
- Biến, phạm vi biến, kiểu dữ liệu, hàm
- Popup boxes
- HTML event handlers
- Javascript nâng cao

Giới thiệu Javascript



Giới thiệu Javascript (tt)

- Javascript
  - Là ngôn ngữ kịch bản, hướng đối tượng, đa nền được thông dịch và thực thi bởi trình duyệt tại client
  - Khác với Java
  - Được hỗ trợ bởi tất cả các trình duyệt phổ biến: IE, Firefox, Chrome, Opera, Safari,…
  - Tăng tính động cho giao diện web: thay đổi cách trình bày, nội dung khi nhận được sự kiện phát sinh từ người dung hay hệ thống
  - Miễn phí

Giới thiệu Javascript (tt)

- Javascript dùng để làm gì?
  - Viết trang web sinh động
  - Kiểm tra dữ liệu trước người dùng nhập vào trước khi gửi về cho server
  - Bắt và xử lý sự kiện
  - Đọc, viết các thành phần HTML
  - Có thể được dùng để phát hiện ra trình duyệt đang được sử dụng để hiển thị trang web
  - Có thể được dùng để tạo cookie phía client

Giới thiệu Javascript (tt)

- Viết trực tiếp vào HTML output stream

```
document.write("<p>This is a paragraph.</p>");
```

- Thay đổi nội dung HTML

```
x=document.getElementById("demo");      // find an element
x.innerHTML="Hello";                    // change the content
```

- Phản ứng khi có sự kiện

```
<button type="button" onclick="myFunction()">Click me! </button>
```

- Thay đổi HTML style

```
document.getElementById("demo").style.color="#ff0000";
```

Giới thiệu Javascript (tt)

- Kiểm tra form input

```
<form name="myForm" action="server.php" method="post"
onsubmit="return validateForm()" >
Your name: <input type="text" name ="yourname">
<input type="submit" value="Submit">
</form>
```

```
function validateForm() {
        var x=document.forms["myForm"]["yourname"].value;
        if (x==null || x=="") {
                alert("Your name must be filled out");
                return false;
        }
}
```

Giới thiệu Javascript (tt)

- Vị trí nhúng Javascript
  - Trong trang HTML:
    - Mã lệnh Javascript được đặt trong thẻ script, có thể nằm trong thẻ <head> hay thẻ <body>

```
<html>
<body>
<h1>My First Web Page</h1>
<script>
<!--
 document.write("<p>" + Date() + "</p>");

//-->
</script>
</body>
</html>
```

Giới thiệu Javascript (tt)

- Vị trí nhúng Javascript (tt)
  - Đặt trong một file riêng (không chứa HTML):
    - Dùng để chứa mã lệnh được sử dụng cho nhiều trang web khác nhau
    - Có phần mở rộng .js
    - Không chứa thẻ <script>
    - Để chèn mã lệnh Javascript từ một file .js, ta thực hiện như sau

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

Giới thiệu Javascript (tt)

- Cú pháp cơ bản
  - Phân biệt chữ hoa và chữ thường: biến, tên hàm,…
  - Dấu “;” dùng để ngăn cách các câu lệnh được viết trên cùng một dòng.
  - Các định danh trong Javascript theo định dạng sau:
    - Kí tự đầu tiên phải là một chữ cái, “_” hay “$”
    - Theo sau có thể là chữ cái, “_”, “$” hay số
  - Khối lệnh nằm trong cặp dấu ngoặc nhọn {}
  - Có 2 cách chú thích
    - Chú thích trên một dòng // this is a single line comment
    - Chú thích trên nhiều dòng /* this is a multi-line comment */

Giới thiệu Javascript (tt)

- Các cách hiển thị dữ liệu với Javascript
  - Viết vào thành phần HTML, sử dụng **innerHTML**
    - document.getElementById("demo").innerHTML = “Hello!”;
  - Viết vào trang HTML, sử dụng **document.write()**
    - document.write(5 + 6);
    - Không nên sử dụng sau khi trang web đã tải xong vì sẽ xóa hết nội dung HTML hiện tại
  - Hiển thị bảng thông báo (alert box), sử dụng **window.alert()**
    - window.alert(5 + 6);
  - Hiển thị ở console của trình duyệt, sử dụng **console.log()**
    - console.log(5 + 6);

Biến

- Biến dùng để chứa dữ liệu
- Khai báo biến:
  - var x;
  - var name;
- Vừa khai báo vừa gán trị cho biến:
  - var x=5;
  - var name="Viet Nam"
- Dùng dấu nháy kép để bao lại giá trị chuỗi khi gán giá trị cho biến
- Nếu một biến được gán trị nhưng chưa được khai báo thì nó sẽ tự động được khai báo như là biến toàn cục:
  - x=5;
  - name="Viet Nam"
- Nếu biến được khai báo lại thì giá trị trước đó vẫn còn

Phạm vi biến

- Cục bộ
  - Biến khai báo trong hàm, trong một khối lệnh
  - Chỉ có thể được truy xuất trong hàm, khối lệnh đó
  - Bị hủy sau khi hàm hay khối lệnh thực thi xong
- Toàn cục
  - Khai báo bên ngoài bất kỳ hàm hay khối lệnh nào
  - Có thể được truy cập từ bất cứ nơi đâu trên trang web
  - Biến chỉ bị hủy khi trang web được đóng lại
  - Nếu một biến được khai báo không dùng từ khóa **var** thì nó sẽ trở thành biến toàn cục dù được khai báo bất cứ nơi đâu

Kiểu dữ liệu

- Số nguyên
- Số thực và số có dấu chấm động
- Luận lý: true, false
- Chuỗi: “Viet Nam”, “I love you”, “\n”, “\t”,…

```
var quote = “I read \”The Vinci Code\” by Dan Brown.”
document.write(quote);
```

- Dùng toán tử **typeof** để xác định kiểu dữ liệu của một biến, các giá trị có thể là “undefined”, “boolean”, “string”, “object”, “function”

```
var quote = “I read \”The Vinci Code\” by Dan Brown.”
document.write(typeof quote); // display “string”
```

Kiểu dữ liệu (tt)

- Đối tượng
  - String
    - http://www.w3schools.com/jsref/jsref_obj_string.asp
  - Date
    - http://www.w3schools.com/jsref/jsref_obj_date.asp
  - Array
    - http://www.w3schools.com/jsref/jsref_obj_array.asp
  - Boolean
  - Math
    - http://www.w3schools.com/jsref/jsref_obj_math.asp
  - RegExp
    - http://www.w3schools.com/jsref/jsref_obj_regexp.asp
  - HTML DOM

String

Property	Description
constructor	Returns the function that created the String object's prototype
length	Returns the length of a string
prototype	Allows you to add properties and methods to an object

Method	Description
charAt()	Returns the character at the specified index
charCodeAt()	Returns the Unicode of the character at the specified index
concat()	Joins two or more strings, and returns a copy of the joined strings
fromCharCode()	Converts Unicode values to characters
indexOf()	Returns the position of the first found occurrence of a specified value in a string
lastIndexOf()	Returns the position of the last found occurrence of a specified value in a string
match()	Searches for a match between a regular expression and a string, and returns the matches
replace()	Searches for a match between a substring (or regular expression) and a string, and replaces the matched substring with a new substring
search()	Searches for a match between a regular expression and a string, and returns the position of the match
slice()	Extracts a part of a string and returns a new string
split()	Splits a string into an array of substrings
substr()	Extracts the characters from a string, beginning at a specified start position, and through the specified number of character
substring()	Extracts the characters from a string, between two specified indices
toLowerCase()	Converts a string to lowercase letters
toUpperCase()	Converts a string to uppercase letters
valueOf()	Returns the primitive value of a String object

String: ví dụ

- myString=“Go Johnny, Go Go Go”

myString.length → 19
myString.charAt(3) → “J”
myString.indexOf(“Go”) → 0
myString.indexOf(“Go”, 3) → 11
myString.substring(3, 9) → “Johnny”
myString.toLowerCase() → “go johnny, go go go”
myString.replace(/Go/, “Up”) → “Up Johnny, Go Go Go”

Array

Property	Description
constructor	Returns the function that created the Array object's prototype
length	Sets or returns the number of elements in an array
prototype	Allows you to add properties and methods to an Array object

Method	Description
concat()	Joins two or more arrays, and returns a copy of the joined arrays
indexOf()	Search the array for an element and returns it's position
join()	Joins all elements of an array into a string
lastIndexOf()	Search the array for an element, starting at the end, and returns it's position
pop()	Removes the last element of an array, and returns that element
push()	Adds new elements to the end of an array, and returns the new length
reverse()	Reverses the order of the elements in an array
shift()	Removes the first element of an array, and returns that element
slice()	Selects a part of an array, and returns the new array
sort()	Sorts the elements of an array
splice()	Adds/Removes elements from an array
toString()	Converts an array to a string, and returns the result
unshift()	Adds new elements to the beginning of an array, and returns the new length
valueOf()	Returns the primitive value of an array

Array: ví dụ

```
<script>
var fruits = [“banana”, “orange”, “apple”, “mango”];
document.write(fruits.sort());
</script>
```

```
<script>
function sortNumber(a, b) {
        return a – b;
}
var n = [“10”, “5”, “40”, “25”, “100”, “1”];
document.write(n.sort(sortNumber));
</script>
```

Date

Method	Description
getDate()	Returns the day of the month (from 1-31)
getDay()	Returns the day of the week (from 0-6)
getFullYear()	Returns the year (four digits)
getHours()	Returns the hour (from 0-23)
getMilliseconds()	Returns the milliseconds (from 0-999)
getMinutes()	Returns the minutes (from 0-59)
getMonth()	Returns the month (from 0-11)
getSeconds()	Returns the seconds (from 0-59)
getTime()	Returns the number of milliseconds since midnight Jan 1, 1970
getTimezoneOffset()	Returns the time difference between GMT and local time, in minutes
getUTCDate()	Returns the day of the month, according to universal time (from 1-31)
getUTCDay()	Returns the day of the week, according to universal time (from 0-6)
getUTCFullYear()	Returns the year, according to universal time (four digits)
getUTCHours()	Returns the hour, according to universal time (from 0-23)
getUTCMilliseconds()	Returns the milliseconds, according to universal time (from 0-999)
getUTCMinutes()	Returns the minutes, according to universal time (from 0-59)
getUTCMonth()	Returns the month, according to universal time (from 0-11)
getUTCSeconds()	Returns the seconds, according to universal time (from 0-59)
parse()	Parses a date string and returns the number of milliseconds since midnight of January 1, 1970
setDate()	Sets the day of the month of a date object
setFullYear()	Sets the year (four digits) of a date object

setHours()	Sets the hour of a date object
setMilliseconds()	Sets the milliseconds of a date object
setMinutes()	Set the minutes of a date object
setMonth()	Sets the month of a date object
setSeconds()	Sets the seconds of a date object
setTime()	Sets a date and time by adding or subtracting a specified number of milliseconds to/from midnight January 1, 1970
setUTCDate()	Sets the day of the month of a date object, according to universal time
setUTCFullYear()	Sets the year of a date object, according to universal time (four digits)
setUTCHours()	Sets the hour of a date object, according to universal time
setUTCMilliseconds()	Sets the milliseconds of a date object, according to universal time
setUTCMinutes()	Set the minutes of a date object, according to universal time
setUTCMonth()	Sets the month of a date object, according to universal time
setUTCSeconds()	Set the seconds of a date object, according to universal time
toDateString()	Converts the date portion of a Date object into a readable string
toISOString()	Returns the date as a string, using the ISO standard
toJSON()	Returns the date as a string, formated as a JSON date
toLocaleDateString()	Returns the date portion of a Date object as a string, using locale conventions
toLocaleTimeString()	Returns the time portion of a Date object as a string, using locale conventions
toLocaleString()	Converts a Date object to a string, using locale conventions
toString()	Converts a Date object to a string
toTimeString()	Converts the time portion of a Date object to a string
toUTCString()	Converts a Date object to a string, according to universal time
UTC()	Returns the number of milliseconds in a date string since midnight of January 1, 1970, according to universal time
valueOf()	Returns the primitive value of a Date object

Date (tt)

Modifier	Description
i	Perform case-insensitive matching
g	Perform a global match (find all matches rather than stopping after the first match)
m	Perform multiline matching

RegExp

Expression	Description
[abc]	Find any character between the brackets
[^abc]	Find any character not between the brackets
[0-9]	Find any digit from 0 to 9
[A-Z]	Find any character from uppercase A to uppercase Z
[a-z]	Find any character from lowercase a to lowercase z
[A-z]	Find any character from uppercase A to lowercase z
[adgk]	Find any character in the given set
[^adgk]	Find any character outside the given set
(red|blue|green)	Find any of the alternatives specified

Quantifier	Description
n+	Matches any string that contains at least one n
n*	Matches any string that contains zero or more occurrences of n
n?	Matches any string that contains zero or one occurrences of n
n{X}	Matches any string that contains a sequence of *X* *n*'s
n{X,Y}	Matches any string that contains a sequence of X to Y *n*'s
n{X,}	Matches any string that contains a sequence of at least X *n*'s
n$	Matches any string with n at the end of it
^n	Matches any string with n at the beginning of it
?=n	Matches any string that is followed by a specific string n
?!n	Matches any string that is not followed by a specific string n

RegExp (tt)

Metacharacter	Description
.	Find a single character, except newline or line terminator
\w	Find a word character
\W	Find a non-word character
\d	Find a digit
\D	Find a non-digit character
\s	Find a whitespace character
\S	Find a non-whitespace character
\b	Find a match at the beginning/end of a word
\B	Find a match not at the beginning/end of a word
\0	Find a NUL character
\n	Find a new line character
\f	Find a form feed character
\r	Find a carriage return character
\t	Find a tab character
\v	Find a vertical tab character
\xxx	Find the character specified by an octal number xxx
\xdd	Find the character specified by a hexadecimal number dd
\uxxxx	Find the Unicode character specified by a hexadecimal number xxxx

RegExp (tt)

RegExp Object Properties

Property	Description
global	Specifies if the "g" modifier is set
ignoreCase	Specifies if the "i" modifier is set
lastIndex	The index at which to start the next match
multiline	Specifies if the "m" modifier is set
source	The text of the RegExp pattern

RegExp Object Methods

Method	Description
compile()	Compiles a regular expression
exec()	Tests for a match in a string. Returns the first match
test()	Tests for a match in a string. Returns true or false

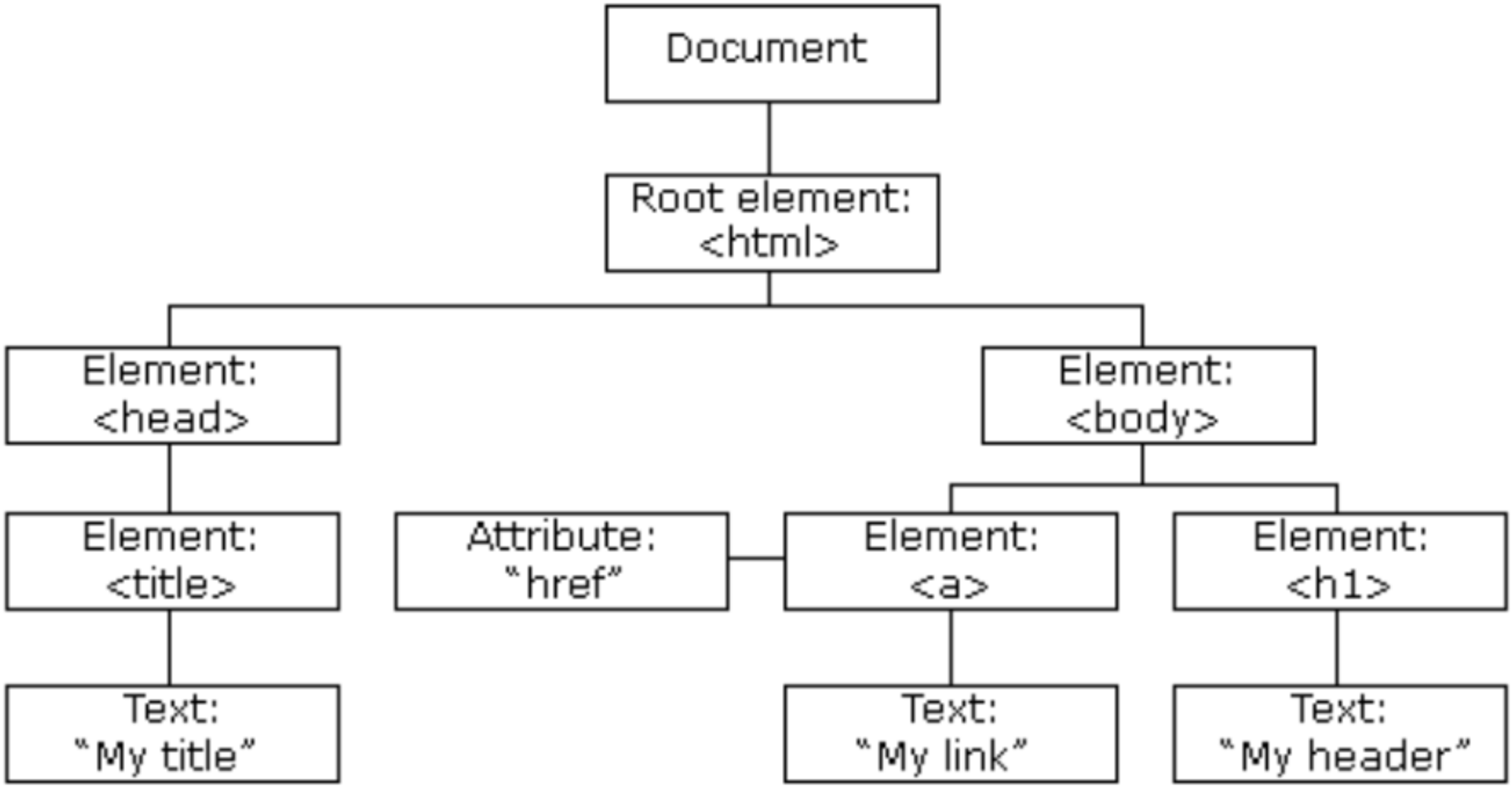
RegExp: ví dụ

```
// match all instances of “at” in a string
var pattern1=/at/g;
// match the first instance of “bat” or “cat”, regardless of case
var pattern2=/[bc]at/i;
```

```
// email address
var RE_EMAIL = /^(\w+\.)*\w+@(\w+\.)+[A-Za-z]+$/;
// date string dd/mm/yyyy
var RE_DATE = /(0[1-9]|[12][0-9]|3[01])[-/.](0[1-9]|1[012])[-/.](19|20)[0-9]{2}/;
// number
var RE_NUM = /\b\d+\b/;
// string has 1 to 10 characters
var RE_STR = /^[A-Za-z]{1,10}$/;
```

HTML DOM

- DOM: Document Object Model



HTML DOM Document Object

Find HTML elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

Changing HTML elements

Method	Description
element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element

Kiểu dữ liệu (tt)

- Được chuyển động

```
var x=10;
x="5"
x=10 + "5" // quy tắc: chuỗi cộng số kết quả luôn là chuỗi
```

- Chuyển chuỗi sang số: *ParseInt()*; *ParseFloat()*, *Number()*
- Chuyển sang kiểu chuỗi: *toString()*, *String()*
- Chuyển sang kiểu Boolean(): *Boolean()*

```
var num1=Number("Viet Nam");  // NaN
var num2=Number("");          // 0
var num3=Number("000011");    // 11
var num4=Number("true");      // 1
```

```
var num=10;
alert(num.toString());        // "10"
alert(num.toString(2));       // "1010"
alert(num.toString(10));      // "10"
alert(num.toString(16));      // "a"
```

```
var value1=10;      alert(String(value1));    // "10"
var value2=true;    alert(String(value2));    // "true"
var value3=null;    alert(String(value3));    // "null"
var value4;         alert(String(value4));    // "undefined"
```

Toán tử

- Toán tử số học

Operator	Description	Example	Result	
+	Addition	x=y+2	x=7	y=5
-	Subtraction	x=y-2	x=3	y=5
*	Multiplication	x=y*2	x=10	y=5
/	Division	x=y/2	x=2.5	y=5
%	Modulus (division remainder)	x=y%2	x=1	y=5
++	Increment	x=++y	x=6	y=6
		x=y++	x=5	y=6
--	Decrement	x=--y	x=4	y=4
		x=y--	x=5	y=4

- Toán tử gán

Operator	Example	Same As	Result
=	x=y		x=5
+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

Toán tử (tt)

- Toán tử so sánh

Operator	Description	Example
==	is equal to	x==8 is false x==5 is true
===	is exactly equal to (value and type)	x===5 is true x==="5" is false
!=	is not equal	x!=8 is true
>	is greater than	x>8 is false
<	is less than	x<8 is true
>=	is greater than or equal to	x>=8 is false
<=	is less than or equal to	x<=8 is true

- Toán tử logic

Operator	Description	Example
&&	and	(x < 10 && y > 1) is true
||	or	(x==5 || y==5) is false
!	not	!(x==y) is true

Ký tự đặc biệt

- Các ký tự đặc biệt có thể được chèn vào chuỗi bằng cách dùng ký tự backslash "\"

Code	Outputs
\'	single quote
\"	double quote
\\	backslash
\n	new line
\r	carriage return
\t	tab
\b	backspace
\f	form feed

```
var txt="We are the so-called \"Vikings\" from the north.";
document.write(txt);
```

Các lệnh điều khiển

- if-else

```
<html>
<body>
<script>
        var date = new Date();
        var time = date.getHours();
        if (time < 10) {
                document.write("<p> Good morning! </p>");
        } else {
                 document.write("<p> Good day! </p>");
        }
</script>
</body>
</html>
```

Các lệnh điều khiển (tt)

- switch-case

```
<html><body>
<script>
        var date = new Date();
        var theDay = date.getDay();
        switch (theDay) {
        case 0:
                document.write("<p>Hello Sunday!</p>"); break;
        case 6:
                document.write("<p>Hello Saturday!</p>"); break;
        default:
                document.write("<p>I'm working!</p>"); break;
        }
</script>
</body></html>
```

Các lệnh điều khiển (tt)

- for

```
<html><body>
<script>
        var person={fname: “John”, lname: “Doe”, age: 25};
        for (x in person) {
                document.write(person[x] + “ ”);
        }

        for (i=1;  i<=6; i++) {
                document.write(“<h” + i + “> This is header ” + i);
                document.write(“</h” + i +“>”);
        }
</script>
</body></html>
```

Các lệnh điều khiển (tt)

- while

```
<html><body>
<script>
        var i=1;
        while (i<=6) {
                document.write("<h" + i + "> This is header " + i);
                document.write("</h" + i +">");
                i=i+1;
        }
</script>
</body></html>
```

Các lệnh điều khiển (tt)

- do-while

```
<html><body>
<script>
        var i=1;
        do {
                document.write("<h" + i + "> This is header " + i);
                document.write("</h" + i +">");
                i=i+1;
        } while (i<=6)
</script>
</body></html>
```

Các lệnh điều khiển (tt)

```
<html><body>
<script type="text/javascript">
        for (i=0;i<=5;i++) {
                if (i==3) {break}
                document.write("<p>Paragraph " + i+ "</p>");
        }
</script>
</body></html>
```

```
<html><body>
<script type="text/javascript">
        for (i=0;i<=5;i++) {
                if (i==3) {continue}
                document.write("<p>Paragraph " + i+ "</p>");
        }
</script>
</body></html>
```

Hàm

- Các hàm có thể được kích hoạt bởi một sự kiện hay một lời gọi hàm
- Các hàm có thể được định nghĩa trong thẻ <head> hay <body> của trang web nhưng phải đảm bảo rằng một hàm đã được nạp trước bất cứ lời gọi nào đến nó
- Hàm ẩn danh (anonymous function)

```
<html><body>
<p id="demo"></p>
<script>
        var x = function (a, b) {return a * b};
        document.getElementById("demo").innerHTML = x(4, 3);
</script>
</body></html>
```

Popup Boxes

- Alert box: thường được sử dụng để đưa một thông điệp nhắc nhở người dùng về một thao tác hay lỗi nào đó.
- Cú pháp: ***alert("some text");***

```
<html><head>
<script>
function disp_alert() {
    alert("Hello! This is how we " + '\n' + " add a line break to an alert box!");
}
</script>
</head>
<body>
<form><input type="button" onclick="disp_alert()"></form>
</body></html>
```

Popup Boxes (tt)

- Confirm box: được sử dụng để yêu cầu người dùng xác nhận hay không một việc gì đó. Confirm box trả về *true* khi người dùng nhấn "OK" và trả về *false* khi người dùng nhấn "Cancel"
- Cú pháp: ***confirm("some text");***

Popup Boxes (tt)

```
<html><head>
<script>
function disp_confirm() {
        var name = confirm("Press a button…");
        if (name == true) {
                document.write("You pressed OK <br>");
        } else {
                 document.write("You pressed Cancel <br>");
        }
}
</script>
</head>
<body>
<form><input type="button" onclick="disp_confirm()" value="Display a
confirm box"></form>
</body></html>
```

Popup Boxes (tt)

- Prompt box: được sử dụng để yêu cầu người dùng nhập vào một giá trị hoặc một thông tin nào đó. Prompt box trả về giá trị người dùng nhập nếu người dùng nhấn "OK" và trả về *null* khi người dùng nhấn "Cancel"
- Cú pháp: ***prompt("some text", "default value");***

Popup Boxes (tt)

```
<html><head>
<script>
function disp_prompt() {
        var name = prompt("Please enter your name…", "");
        if (name != true || name != "") {
                document.write("Your name is " + name + "<br>");
        }
}
</script>
</head>
<body>
<form><input type="button" onclick="disp_prompt()" value="Display a
prompt box"></form>
</body></html>
```

HTML Event Handlers		
Event Handler	**Elements Supported**	**Description**
onblur	a, area, button, input, label, select, textarea	the element lost the focus
onchange	input, select, textarea	the element value was changed
onclick	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer button was clicked
ondblclick	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer button was double clicked
onfocus	a, area, button, input, label, select, textarea	the element received the focus
onkeydown	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a key was pressed down
onkeypress	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a key was pressed and released
onkeyup	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a key was released
onload	frameset	all the frames have been loaded

onload	body	the document has been loaded
onmousedown	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer button was pressed down
onmousemove	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer was moved within
onmouseout	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer was moved away
onmouseover	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer was moved onto
onmouseup	All elements except applet, base, basefont, bdo, br, font, frame, frameset, head, html, iframe, isindex, meta, param, script, style, title	a pointer button was released
onreset	form	the form was reset
onselect	input, textarea	some text was selected
onsubmit	form	the form was submitted
onunload	frameset	all the frames have been removed
onunload	body	the document has been removed

HTML event handlers (tt)

```
<html>
<head>
<script>
function displayDate() {
        document.getElementById("demo").innerHTML=Date();
}
</script>
</head>
<body>
<h1>My First Web Page</h1>
<p id="demo"></p>
<button type="button" onclick="displayDate()">Display Date</button>
</body>
</html>
```

Nâng cao

- Thông tin về trình duyệt

```
<div id="example"></div>

<script>

txt = "<p>Browser CodeName: " + navigator.appCodeName + "</p>";
txt+= "<p>Browser Name: " + navigator.appName + "</p>";
txt+= "<p>Browser Version: " + navigator.appVersion + "</p>";
txt+= "<p>Cookies Enabled: " + navigator.cookieEnabled + "</p>";
txt+= "<p>Platform: " + navigator.platform + "</p>";
txt+= "<p>User-agent header: " + navigator.userAgent + "</p>";

document.getElementById("example").innerHTML=txt;

</script>
```

Nâng cao (tt)

- Làm việc với cookie

```
function setCookie(c_name,value,exdays) {
        var exdate=new Date();
        exdate.setDate(exdate.getDate() + exdays);
        var c_value=escape(value) + ((exdays==null) ? "" : ";
expires="+exdate.toUTCString());
        document.cookie=c_name + "=" + c_value;
}
```

Nâng cao (tt)

- Làm việc với cookie (tt)

```
function getCookie(c_name) {
var i, x, y, ARRcookies=document.cookie.split(";");
for (i=0; i<ARRcookies.length; i++) {
        x=ARRcookies[i].substr(0, ARRcookies[i].indexOf("="));
        y=ARRcookies[i].substr(ARRcookies[i].indexOf("=")+1);
        x=x.replace(/^\s+|\s+$/g,"");
        if (x==c_name) {
                return unescape(y);
        }
}
}
```

Nâng cao (tt)

- Làm việc với cookie (tt)

```
function checkCookie() {
var username=getCookie("username");
if (username!=null && username!="") {
        alert("Welcome again " + username);
} else {
        username=prompt("Please enter your name:","");
        if (username!=null && username!="") {
                setCookie("username",username,365);
        }
}
}
```

Nâng cao (tt)

- Sự kiện thời gian
  - setInterval("myFunction()", 1000)
    - Cứ sau 1 giây thì thực hiện myFunction(), lặp lại
  - setTimeout ("myFunction()", 1000)
    - Sau 1 giây thì thực hiện myFunction(), chỉ thực hiện 1 lần
  - clearInterval()
    - Dừng thực hiện hàm được gọi bởi setInterval()
  - clearTimeout()
    - Dừng thực hiện hàm được gọi bởi setTimeout()

```
<!DOCTYPE html>
<html><body>
<button onclick="myFunction()">Try it</button>
<button onclick="myStopFunction()">Stop time</button>

<script>
    var myVar;
    function myFunction() {
        myVar=setInterval("myTimer()",1000);
    }
    function myTimer() {
        var d=new Date();
        var t=d.toLocaleTimeString();
        document.getElementById("demo").innerHTML=t;
    }
    function myStopFunction() {
        clearInterval(myVar);
    }
</script>

<p id="demo"></p>
</body></html>
```

Nâng cao (tt)

- Bắt và xử lý lỗi

```
<!DOCTYPE html><html><head>
<script>
function message() {
        try {
                adddlert("Welcome guest!");
        } catch (error) {
                alert(error);
        }
}
</script>
</head><body>
<input type="button" value="View message" onclick="message()">
</body></html>
```

HẾT