

Chuyên đề CSDL & WEB

Ngôn ngữ Client Script (Javascript)

Lương Vĩ Minh - Nguyễn Trường Sơn

Khoa CNTT – ĐH.KHTN

Nội dung

- Ví dụ mở đầu
- Ngôn ngữ Client Script
- DHTML
- Javascript
- VBScript

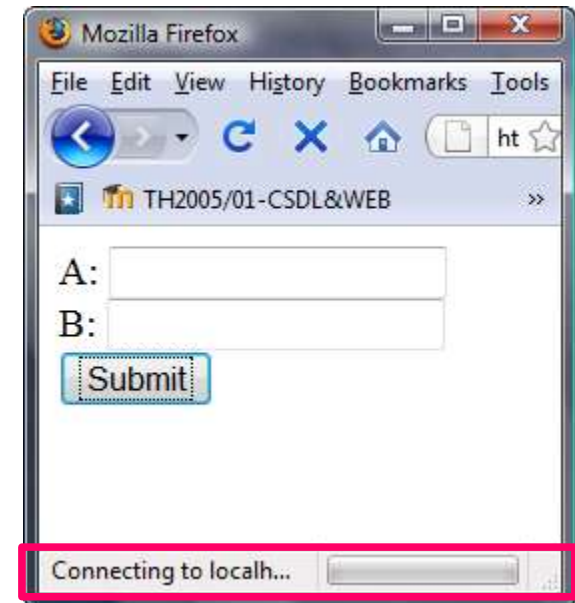
Nội dung

- Ví dụ mở đầu
- Ngôn ngữ Client Script
- DHTML
- Javascript
- VBScript

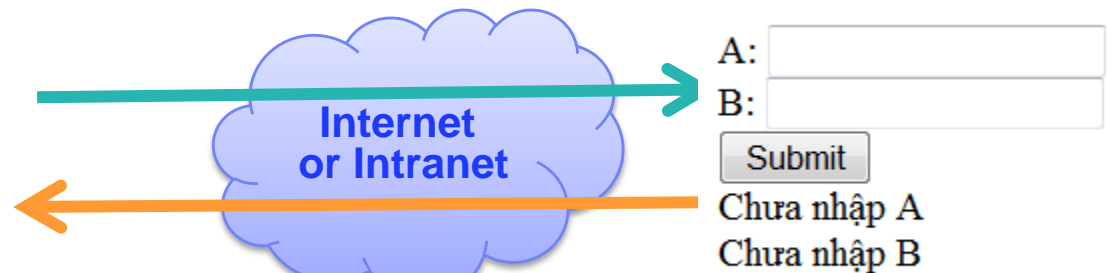
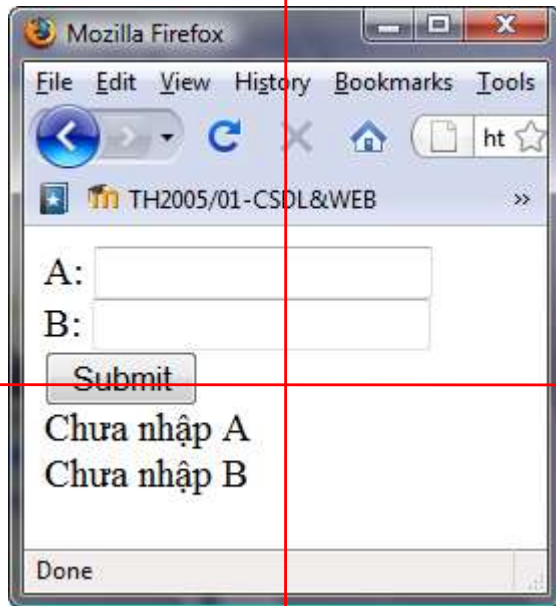
Ví dụ mở đầu

ViDuMoDau.php

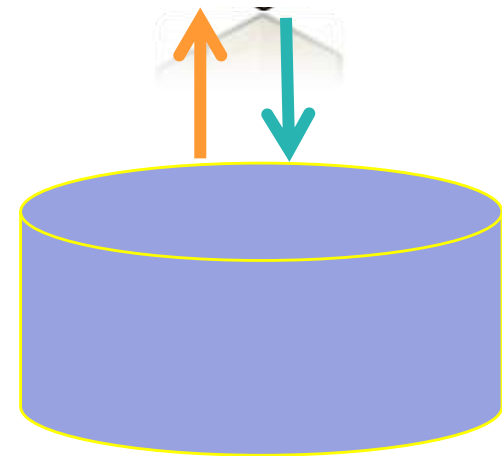
```
<form action="Bai7_ViDuMoDau.php" method="get">
  A: <input type="text" name="txtA"><br>
  B: <input type="text" name="txtB"><br>
  <input type="submit" value="Submit">
  <?php
    if (isset($_GET['txtA']) && $_GET['txtA']=='')
    {
      echo "<br/> Chưa nhập A";
    }
    if (isset($_GET['txtB']) && $_GET['txtB']=='')
    {
      echo "<br/> Chưa nhập B";
    }
  ?>
</form>
```



Ví dụ mở đầu - Tình huống



**Vấn đề kiểm tra
dữ liệu ?**



Ví dụ mở đầu - Nhận xét

- Việc kiểm tra dữ liệu được thực hiện ở Server → Mất thời gian chờ
 - Việc kiểm tra có thể thực hiện được ở Client vì tại client cũng có đủ điều kiện để kiểm tra.
 - Việc kiểm tra thực hiện ở Client → Sẽ không mất thời gian chờ
- ➔ Cần một ngôn ngữ để thực thi ở trình duyệt.

Ví dụ mở đầu - Nhận xét

- Nhu cầu:
 - Xử lý các tương tác với người dùng: các sự kiện chuột
 - Xử lý các hiệu ứng đồ họa : tạo các menu, hiệu ứng hình ảnh ...
 - Thực hiện một số xử lý chức năng ở trình duyệt trước khi submit: kiểm tra dữ liệu, ...
 - Thực hiện một số chức năng: bộ gõ tiếng việt, ...
- Phương pháp:
 - Dùng ngôn ngữ server-side → **không hợp lý.**
 - Cần một ngôn ngữ thực thi ở trình duyệt



Client Script

Nội dung

- Ví dụ mở đầu
- Ngôn ngữ Client Script
- DHTML
- Javascript
- VBScript

Ngôn ngữ Script – Giới thiệu

- Là ngôn ngữ dạng thông dịch.
- Giúp trang web có tính tương tác với người dùng
- Các ngôn ngữ script thông dụng
 - Javascript (Netscape)
 - VBScript (Microsoft)
- Trình duyệt web đóng vai trò là trình biên dịch của ngôn ngữ.

Ngôn ngữ Script - Ứng dụng ClientSide và ServerSide

■ Ứng dụng Client-Side:

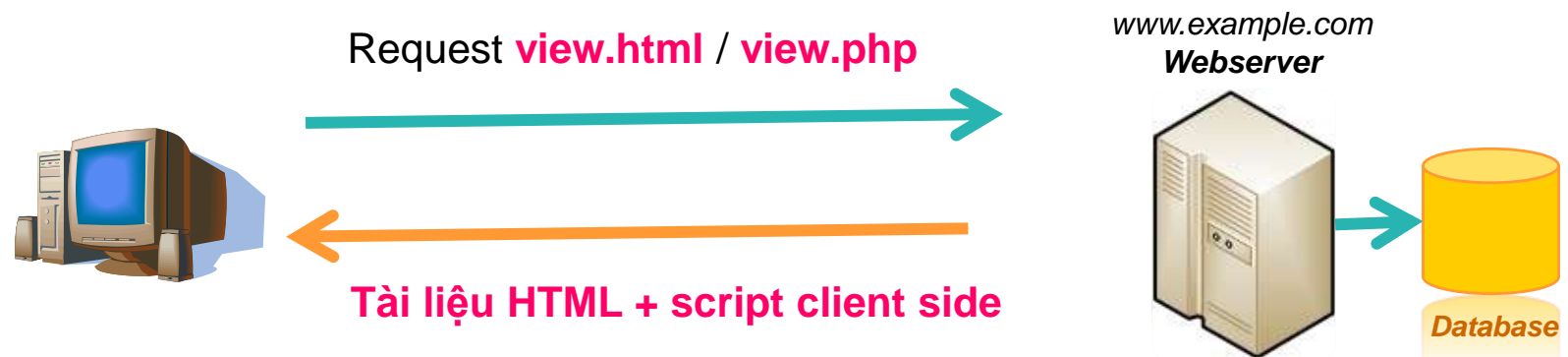
- Thực hiện tại **Browser** (Nescape Navigator, IE, Firefox, Safari, ...)
- Script tại Client-Side (Thực hiện các tương tác với người dùng, thay đổi cấu trúc trang web, kiểm tra dữ liệu được nhập vào của người dùng, ...)

■ Ứng dụng Server-Side:

- Thực hiện tại **WebServer** (IIS, Netscape Enterprise Server,)
- Script tại Server-Side (kết nối CSDL, chia sẻ thông tin giữa các người duyệt web, truy cập hệ thống file trên server, ...)

Ngôn ngữ Script – Quá trình thực hiện script tại server

- Quá trình thực hiện ứng dụng Server-Side gồm 2 giai đoạn:
 - Tạo trang Web có chứa cả HTML + Script Client-Side và Script Server-Side
 - Khi Client browser yêu cầu thực hiện, server (run-time engine) sẽ thực hiện các lệnh Server-side Scripts và **trả trang Web HTML + Script Client-Side** về browser



Ngôn ngữ script – Nhúng Script vào trang web

- Định nghĩa script trực tiếp

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

- Nhúng script từ 1 file khác

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

- Định nghĩa script trực tiếp

```
<script type="text/vbscript">  
    <!--  
        ' Lệnh VBScript  
    -->  
</script>
```

- Nhúng script từ 1 file khác

```
<script src="file.vbs"></script>
```

Ngôn ngữ script – Vị trí đặt script trong HTML

- Đặt giữa tag `<head>` và `</head>`: đoạn script sẽ thực thi ngay khi trang web được mở.
- Đặt giữa tag `<body>` và `</body>`: Đoạn 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 script không hạn chế.

Nội dung

- Ví dụ mở đầu
- Ngôn ngữ Client Script
- DHTML
- Javascript
- VBScript

DHTML

- **DHTML** = Dynamic HyperText Markup Language
- DHTML = HTML + CSS + ClientScript + HTML DOM
- Tích hợp các tính năng của các trình duyệt thế hệ thứ 4 (IEv5, Netscape 4, Firefox 2.0+, Opera 7.0, ...)

Nội dung

- Ví dụ mở đầu
- Ngôn ngữ Client Script
- DHTML
- Javascript
- VBScript

Một số ví dụ

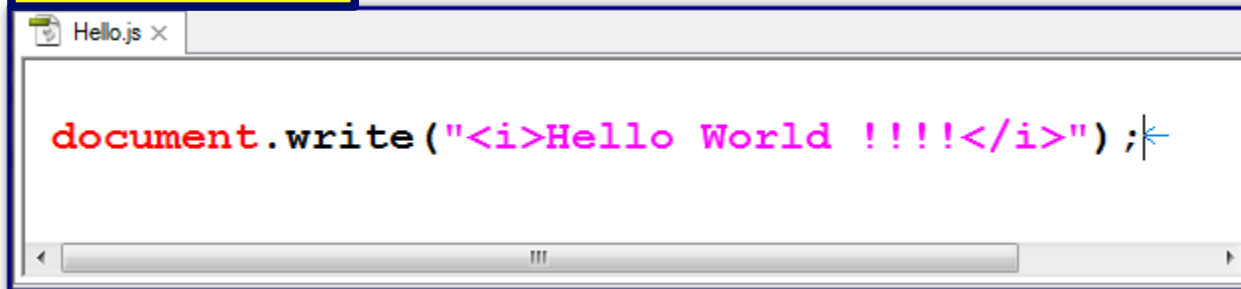
HelloWorld.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Hello World</title>
</head>
<body>
  <script type="text/javascript" language="javascript">
    document.write("<i>Hello World !!!!</i>");
  </script>
</body>
</html>
```



Một số ví dụ

Hello.js



```
document.write("<i>Hello World !!!!</i>");
```

HelloWorld.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Hello World</title>
</head>
<body>
  <script src="Hello.js"> </script>
</body>
</html>
```



Một số ví dụ



```
<form method="get" action="x1DangKyTaiKhoan.php" onsubmit="return CheckForm();" >
```

Tên đăng nhập	<input type="text"/>
Mật khẩu	<input type="password"/>
Nhập lại mật khẩu	<input type="password"/>
<input type="button" value="Submit"/>	

Done



```
<html>
<body>
<?php
    echo $_GET["TenDangNhap"] . " " $_GET["MatKhau"] ;
?>
</body>
</html>
```

```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Index</title>
<script type="text/javascript" language="javascript">
function CheckForm()
{
    var tendangnhap = document.getElementsByName("TenDangNhap") [0].value;
    var matkhau = document.getElementsByName("MatKhau") [0].value;
    if(tendangnhap == "" || matkhau == "")
    {
        alert ("Chưa nhập tên đăng nhập hoặc mật khẩu !!!");
        return false;
    }
    return true;
}
</script>
</head>
```

Javascript – Nội dung

- Cú pháp và quy ước
- Một số cấu trúc lập trình cơ bản
 - Khai báo biến
 - Kiểu dữ liệu
 - Toán tử
 - Một số đối tượng dữ liệu
 - Cấu trúc điều khiển
 - Hàm

Javascript – Nội dung

- **Cú pháp và quy ước**
- Một số cấu trúc lập trình cơ bản
 - Khai báo biến
 - Kiểu dữ liệu
 - Toán tử
 - Một số đối tượng dữ liệu
 - Cấu trúc điều khiển
 - Hàm

Javascript – Cú pháp và quy ước

- Javascript **phân biệt** chữ hoa – chữ thường
 - Ví dụ : Hai biến Java, java là khác nhau
- Tất cả các câu lệnh javascript đều cách nhau bởi dấu ";"
- **Không phân biệt** khoảng trắng, Tab, xuống dòng trong câu lệnh.
- Chuỗi và dấu nháy
 - Chuỗi trong javascript được đặt trong cặp nháy đơn (' ') hoặc nháy kép (" ")
 - Ví dụ :


```
<input type="button" value="Click Me!" onclick="alert('Hello');"/>
<input type="button" value="Click Me!" onclick="alert("Hello");"/>
```

Javascript – Cú pháp và quy ước

■ Ký tự '\':

\b : Backspace	\' : Dấu nháy đơn	\t : tab
\f : Form feed	\" : Dấu nháy kép	
\n : New line	\r : carriage return	

Ví dụ : 'You did\'t get that done'

■ Các loại dấu ngoặc :

{ }	Đánh dấu khối lệnh
[]	Sử dụng trong cấu trúc Mảng
()	Sử dụng trong hàm, thuộc tính đối tượng

Javascript – Cú pháp và quy ước

- **Ghi chú** : Theo cú pháp ghi chú của C++

// Đây là ghi chú

/* Đây là ghi
chú nhiều dòng*/

- **Tên biến và hàm** :

- Bắt đầu bằng Ký tự (A..Z, a..z), _, \$
- **Không** được bắt đầu bằng ký số (0..9)
- **Không** có khoảng trắng giữa tên (biến hoặc hàm)
- **Không** được đặt tên trùng từ khóa
- Ví dụ :

x x13 _money \$money

add_two_num(....) addTwoNum(.....)

Javascript – Danh sách từ khóa

abstract	delete	innerWidth	Packages	status
alert	do	instanceof	pageXOffset	statusbar
arguments	document	int	pageYOffset	stop
Array	double	interface	parent	String
blur	else	isFinite	parseFloat	super
boolean	enum	isNaN	parseInt	switch
Boolean	escape	java	personalbar	synchronized
break	eval	length	print	this
byte	export	location	private	throw
callee	extends	locationbar	prompt	throws
caller	final	long	protected	toolbar
captureEvents	finally	Math	prototype	top
case	find	menubar	public	toString
catch	float	moveBy	RegExp	transient
char	focus	moveTo	releaseEvents	try
class	for	name	resizeBy	typeof
clearInterval	frames	NaN	resizeTo	unescape
clearTimeout	Function	native	return	unwatch
close	function	netscape	routeEvent	valueOf
closed	goto	new	scroll	var
confirm	history	null	scrollbars	void
const	home	Number	scrollBy	watch
constructor	if	Object	scrollTo	while
continue	implements	open	self	window
Date	import	opener	setInterval	with
debugger	in	outerHeight	setTimeout	FALSE
default	Infinity	outerWidth	short	TRUE
defaultStatus	innerHeight	package	static	

Javascript – Khai báo biến

- Khai báo biến bằng từ khóa **var**
- Biến phải được khai báo trước khi sử dụng
- Thật sự tồn tại khi bắt đầu gán giá trị cho biến lần đầu tiên
- Ví dụ:

```
var x ;
```

```
var y, sum ;
```

```
var x = 1, y = -10, sum = 0;
```

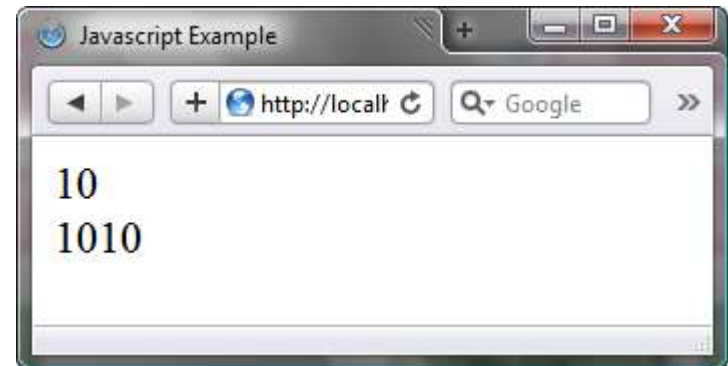
Javascript – Kiểu dữ liệu

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>
function	<code>var add = new function("x", "y", "return(x+y)");</code> <code>add(2, 3);</code>	<i>functionName</i> = new function ([argname1, [... argnameN,]] body);

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

Javascript – Kiểu dữ liệu

```
var a = 10;  
document.write(a + "<br>");  
a = a + "10" ;  
document.write(a);
```



Javascript – Toán tử

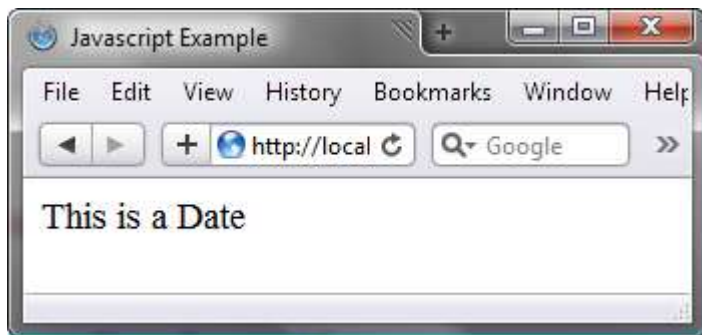
Loại	Toán tử
Toán học	+ - * / % ++ --
So sánh	< > <= >= != ==
Luận lý	&& ?: ,
Xử lý bit	~ << >> >>> & (and) (or) ^ (xor)
Gán	= += -= *= /= %= >>= <<= &= = ^=

Javascript – Một số đối tượng dữ liệu

- String Object
- Number Object
- Date Object
- Math Object
- Array Object
- ActiveX Object

Javascript - thuộc tính **constructor**

- Sử dụng để kiểm tra kiểu dữ liệu của 1 đối tượng
- Cú pháp:
 - *object*.**constructor**



Javascript - thuộc tính **prototype**

- Cho phép thêm một **thuộc tính** hoặc một **phương thức** vào một lớp đối tượng.



```
<body>
<script type="text/javascript">
    function fBlue ()
    {
        return this.fontcolor("blue");
    }
    function fH1 ()
    {
        return "<h1>" + this + "</h1>";
    }
    String.prototype.Blue = fBlue;
    String.prototype.H1 = fH1;
    var s = new String("Hello World !!!");
    var t = new String("Hello !!!");
    document.writeln(s.H1());
    document.writeln(t.Blue());
</script>
</body>
```


Javascript - String. Thuộc tính

- `length` : chiều dài của chuỗi
- `prototype`, `constructor`
- Nối kết các chuỗi bằng toán tử `+`

```
<body>
<script type="text/javascript">
    var s = new String("Hello World !!!");
    document.writeln(s.length);
    document.writeln(s.toUpperCase());
    document.writeln(s.toLowerCase());
</script>
</body>
```

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

Javascript – String . Phương thức

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 = "ABCDEFGHJKLMN" 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>

Javascript – String . Phương thức

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); // < A name="fish_infor">Information about fish</pre>

Javascript – Number

- Kiểu dữ liệu số nguyên, số thực
- Bắt đầu bằng ký số "0" : Số nguyên hệ bát phân
- Bắt đầu bằng "0x" : Số nguyên hệ thập lục phân
- VD : Cho biết giá trị thập phân tương ứng của các number sau :
 - 125 = ?
 - 010 = ? 014 = ? 028 = ?
 - 0xFF = ? 0x3.12 = ?

Javascript – Number . **Thuộc tính**

- constructor
- prototype
- MAX_VALUE (khoảng = 1.79E+308)
- MIN_VALUE (khoảng = 5.00E-324)
- NaN : đại diện cho 1 giá trị không phải là 1 số
- NEGATIVE_INFINITY: đại diện cho 1 giá trị < MIN_VALUE
- POSITIVE_INFINITY: đại diện cho 1 giá trị > MAX_VALUE

Javascript – Number . **Phương thức**

- **toExponential**

numObj.**toExponential**(*[fractionDigits]*)

- **toFixed**

numObj.**toFixed**(*[fractionDigits]*)

- **toPrecision**

numObj.**toPrecision** (*[precision]*)

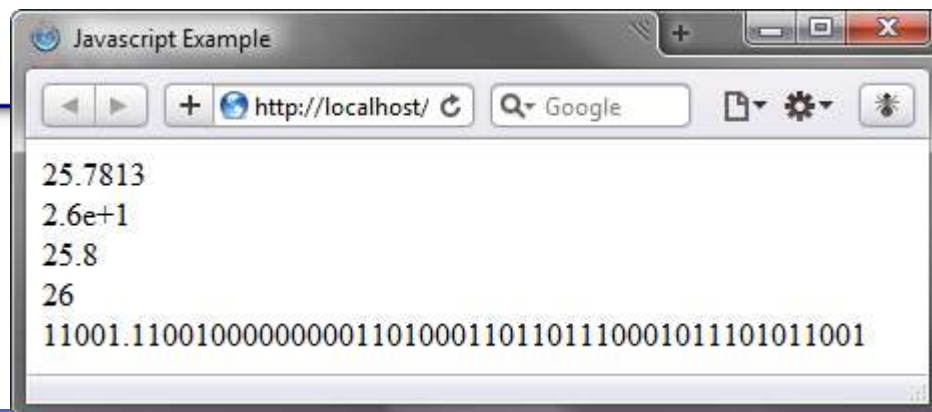
- **toString**

objectname.**toString**(*[radix]*)

radix : [2; 8; 10; 16]

Javascript – Number . Phương thức . Ví dụ

```
<body>
<script type="text/javascript">
    var nNumber = new Number();
    nNumber = 25.7813;
    document.write(nNumber + "<br/>");
    document.write(nNumber.toExponential(1) + "<br/>");
    document.write(nNumber.toFixed(1) + "<br/>");
    document.write(nNumber.toPrecision(2) + "<br/>");
    document.write(nNumber.toString(2) + "<br/>");
</script>
</body>
```



Javascript - Một số hàm chuyển kiểu giá trị

▪ Chuyển chuỗi thành số

parseInt();

parseInt("42")

// result = 42

parseInt("42.33")

// result = 42

parseFloat("42.33")

// result = 42.33

3 + 3 + parseInt("3")

// result = 9

parseFloat();

▪ Chuyển số thành chuỗi (auto):

3 + "3"

// result = "33"

3 + 3 + "3"

// result = "63"

("" + 2500)

// result = "2500"

("" + 2500).length

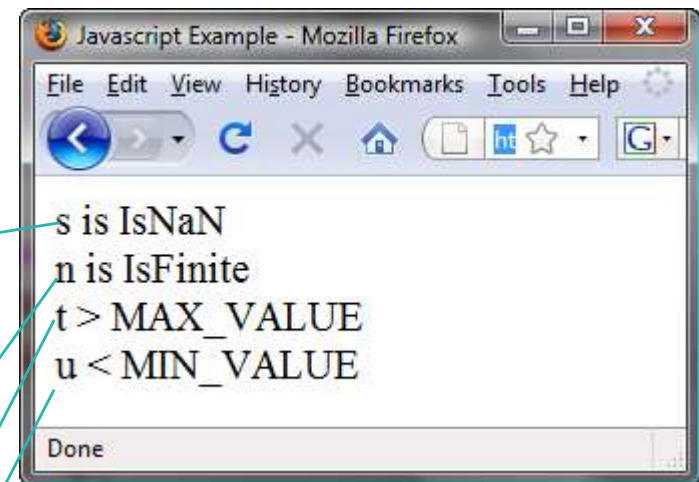
// result = 4

▪ isNaN(**number**) → true nếu **number** là NaN

▪ isFinite(**number**) → true nếu **number** khác NaN, NEGATIVE_INFINITY, POSITIVE_INFINITY

IsNaN và IsFinite

```
<body>
<script type="text/javascript">
    var n = new Number(25.7813);
    var s = new String("Text");
    var t = 10 / 0;
    var u = -10 / 0;
    if (isNaN(n))
        document.write("n is IsNaN <br/>");
    if (isNaN(s))
        document.write("s is IsNaN <br/>");
    if (isNaN(t))
        document.write("t is IsNaN <br/>");
    if (isFinite(t))
        document.write("t is IsFinite <br/>");
    if (isFinite(n))
        document.write("n is IsFinite <br/>");
    if (isFinite(s))
        document.write("s is IsFinite <br/>");
    if (t > Number.MAX_VALUE)
        document.write("t > MAX_VALUE <br/>");
    if (u < Number.MIN_VALUE)
        document.write("u < MIN_VALUE <br/>");
</script>
</body>
```



Javascript - Math

Thuộc tính :

- E
- LN2
- LN10
- LOG2E
- LOG10E
- PI
- SQRT1_2
- SQRT2

Phương thức :

- abs
- log
- acos
- max
- asin
- min
- atan
- pow
- atan2
- random
- ceil
- round
- cos
- sin
- exp
- sqrt
- floor
- tan

Javascript – Date . Thuộc tính

- **var** *dateObj* = **new Date** ()
- **var** *dateObj* = **new Date** (*dateVal*)
- **var** *dateObj* = **new Date** (*year*, *month*,
date [, *hours* [, *minutes* [, *seconds* [, *ms*]]]])
 - month = {0, 1, ..., 11}
 - date = {1, 2, ..., 31}
 - year = {1900, 1901, ...}

Javascript – Date . Phương thức

- getDate
- getDay
- getFullYear
- getHours
- getMilliseconds
- getMinutes
- getMonth
- getSeconds
- getTime
- getTimezoneOffset

- getUTCDate
- getUTCDay
- getUTCFullYear
- getUTCHours
- getUTCMilliseconds
- getUTCMinutes
- getUTCMonth
- getUTCSeconds
- getVarDate
- getYear

- setUTCMonth
- setUTCSeconds
- setYear

- setDate
- setFullYear
- setHours
- setMilliseconds
- setMinutes
- setMonth
- setSeconds
- setTime
- setUTCDate
- setUTCFullYear
- setUTCHours
- setUTCMilliseconds
- setUTCMinutes

- parse
- toString
- toGMTString
- toLocaleDateString
- toLocaleString
- toLocaleTimeString
- toString
- toTimeString
- toUTCString
- UTC

Javascript - Array

- `var arrayObj = new Array()`
- `var arrayObj = new Array([size])`
- `var arrayObj = new Array([element0[, element1[, ...[, elementN]]])`
- `var arrayObj = new Array(Array)`

- Chỉ số mảng bắt đầu bằng chỉ mục 0

- | | | |
|---------------|-----------|------------|
| ■ length | ■ concat | ■ slice |
| ■ constructor | ■ join | ■ sort |
| ■ prototype | ■ pop | ■ splice |
| | ■ push | ■ toString |
| | ■ reverse | ■ unshift |
| | ■ shift | |

Javascript - Array

```
var USStates = new Array(51);           // nodaata in array
USStates[0] = "Alabama";                // phần tử đầu
USStates[1] = "Alaska";
USStates[2] = "Arizona";
USStates[3] = "Arkansas";
...
USStates[50] = "Wyoming";              // phần tử cuối

Rows = USStates.length;                // Rows = 51
var myState = USStates[3];
```

Javascript – Array . Phương thức

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>

Javascript – Array nhiều chiều

■ Ví dụ về mảng 2 chiều

```
var i,j;

var MyArray = new Array(2);

for (i=0; i<MyArray.length; i++)
    MyArray[i] = new Array(3);

MyArray[0][0] = "Ryan Dias";  MyArray[0][1] = 1;  MyArray[0][2] = "Photoshop";
MyArray[1][0] = "Mike Donne";  MyArray[1][1] = 2;  MyArray[1][2] = ".NET";

for (i=1; i>=0; i--)
    for (j=2; j>=0; j--)
        document.write(" " + MyArray[i][j]);
```



■ JavaScript hỗ trợ đến 60 chiều

Javascript – ActiveX Object

- **Cú pháp :**

```
newObj = new ActiveXObject(servername.typename[, location])
Obj = GetObject([pathname] [, class])
```

- **ActiveX - Dictionary:**

```
var y = new ActiveXObject("Scripting.Dictionary");
y.add ("a", "test");
if (y.Exists("a"))
    document.write("true");
```

- **ActiveX - FileSystemObject :**

```
var fso = new ActiveXObject("Scripting.FileSystemObject");
var a = fso.CreateTextFile("c:\\testfile.txt", true);
a.WriteLine("This is a test.");
a.Close();
```

Javascript – Nội dung

- Cú pháp và quy ước
- Một số cấu trúc lập trình cơ bản
 - Khai báo biến
 - Kiểu dữ liệu
 - Toán tử
 - Một số đối tượng dữ liệu
 - Cấu trúc điều khiển
 - Hàm

Javascript – Cấu trúc điều khiển

- Điều kiện `if`
- Điều khiển `switch`
- Vòng lặp `for`
- Vòng lặp `while`
- Vòng lặp `do.. While`
- Vòng lặp `for in`

- Từ khóa : `break`, `continue`

Javascript – 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;
- if (x == 5) z = 7; y = 42; else z = 19;

Javascript – Điều khiển switch

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

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")
}
```

Javascript – 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;  
}
```

Javascript – Vòng lặp while

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

■ Ví dụ:

```
var i = 9, total = 0;
```

```
while (i < 10)
```

```
{
```

```
    total += i * 3 + 5;
```

```
    i = i + 5;
```

```
}
```


Javascript – 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);
```

Javascript – Vòng lặp for ... in

```
for (variable in [object | array])  
{  
    statements  
}
```

- Ví dụ:

```
var a = new Array(2);  
a[0] = 10; a[1] = 20;  
var e, i = 0, b = new Array(2);  
for (e in a) {  
    b[i] = e / 10 - 1;  
    i++;  
}
```

Javascript – Nội dung

- Cú pháp và quy ước
- Một số cấu trúc lập trình cơ bản
 - Khai báo biến
 - Kiểu dữ liệu
 - Toán tử
 - Một số đối tượng dữ liệu
 - Cấu trúc điều khiển
 - Hàm

Javascript - Hàm

```
function functionName ([parameter1] ... [,parameterN])  
{  
    statement[s] ;  
}
```

```
function functionName ([parameter1] ... [,parameterN])  
{  
    statement[s] ;  
    return ... ;  
}
```

Javascript - Hàm

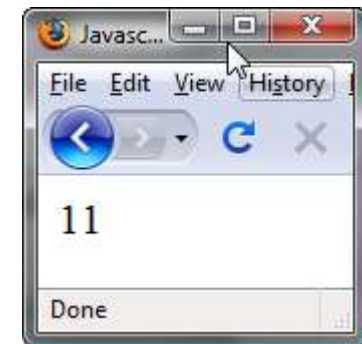
function.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Javascript Example</title>
  <script src="lib.js"> </script>
</head>
<body>

<script type="text/javascript">
  document.write(TinhTong2So(5, 6));
</script>
</body>
</html>
```

lib.js

```
function TinhTong2So(x, y)
{
  return x + y;
}
```



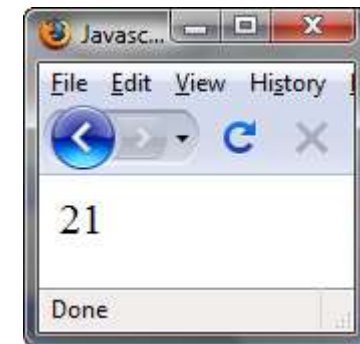
Javascript - Hàm

function.html

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Javascript Example</title>
  <script src="lib.js"> </script>
</head>
<body>
<script type="text/javascript">
  function TinhTong2So(x, y)
  {
    return x + y + 10;
  }
  document.write(TinhTong2So(5, 6));
</script>
</body>
</html>
```

lib.js

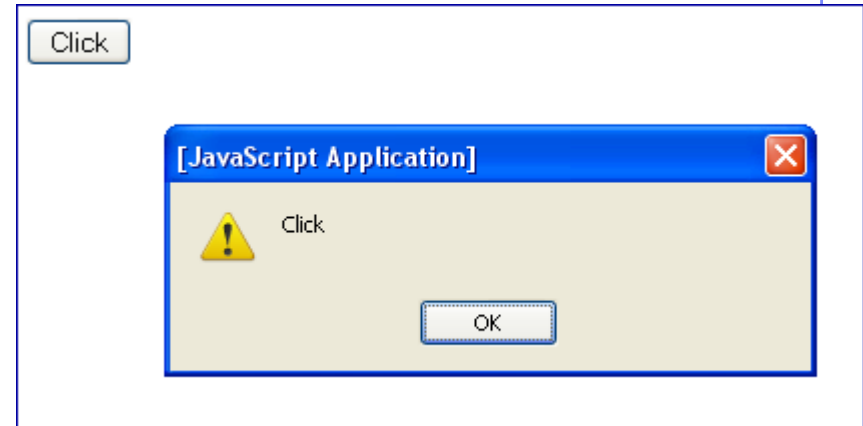
```
function TinhTong2So(x, y)
{
  return x + y;
}
```



Javascript – với các sự kiện

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
    <title>Untitled</title>
    <script type="text/javascript">
        function ShowMsg(s)
        {
            alert(s);
        }
    </script>
</head>
<body>
    <input type="button" value="Click" onclick="ShowMsg(this.value);">

</body>
</html>
```



Nội dung

- DHTML
- Ngôn ngữ Script
- Javascript
- VBScript

VBScript - Giới thiệu

- **VBScript** là ngôn ngữ lập trình kiểu **thông dịch** dành cho Web. Khi đoạn mã **VBScript** được chèn vào trang HTML thì trình duyệt Web sẽ thông dịch và thực hiện
- **VBScript** do **Microsoft** phát triển. Cú pháp của nó gần giống như ngôn ngữ Visual Basic.
- **VBScript** là ngôn ngữ script được **viết phía client** và nó chỉ được hỗ trợ tốt nhất trên trình duyệt Internet Explorer của Microsoft.

VBScript - Biến, kiểu dữ liệu

- Khai báo biến thông qua từ khóa **dim**
 - Dim x, y, z
 - Dim a(10) 'khai báo 1 mảng 1 chiều có 10 phần tử
 - Dim b(5,10) 'khai báo mảng 2 chiều
 - **Redim** a(20) 'khai báo lại mảng a, tăng thêm 10 phần tử (vẫn giữ lại giá trị 10 phần tử đầu)
- Kiểu dữ liệu
 - Khi khai báo, biến chỉ có 1 kiểu là **variant**
 - Các kiểu dữ liệu con (**number**, **string**, **boolean**, **object**, ...) sẽ được gán cho biến khi sử dụng

VBScript - Cấu trúc điều kiện

```
If <biểu thức điều kiện> Then <Lệnh> End if
```

```
If <biểu thức điều kiện> Then  
    ....  
Else  
    ...  
End If
```

```
Select Case <Tên biến>  
    Case <giá trị 1>  
        ....  
    Case <giá trị i>  
        ....  
    Case Else  
        .....  
End Select
```

VBScript - Cấu trúc lặp

```
Do While <Biểu thức điều kiện>  
    ....  
Loop
```

```
Do  
    ....  
Loop While <Biểu thức điều kiện>
```

```
While <Biểu thức điều kiện>  
    .....  
WEnd
```

VBScript - Cấu trúc lặp

```
For <biến đếm>=<Chỉ số đầu> To <Chỉ số cuối>  
    .....  
Next
```

```
For <biến đếm>=<Chỉ số đầu> To <Chỉ số cuối> Step <Bước nhảy>  
    .....  
Next
```

```
For Each <biến> in <biến collection>  
    ...  
Next
```

VBScript - Hàm , thủ tục

```
▪ Function <tên hàm> (<danh sách các tham số>)  
    .....  
  
End Function
```

```
▪ Sub <Tên thủ tục> (<danh sách các tham số>)  
    .....  
  
End Sub
```

VBScript - Hàm , thủ tục

- Gọi hàm

Dim name = **findname**(agr1,...)

- Gọi thủ tục

Call MyProc(arg1,arg2,...)

MyProc arg1,arg2