



Phát triển ứng dụng web 2

Đại Học Sài Gòn – Khoa CNTT

Javascript cơ bản

1 - Javascript cơ bản

GV: Phan Thị Kim Loan

Kiến thức đã học

1. Giới thiệu CSS
2. Định nghĩa Style
3. Phân loại CSS
4. Phạm vi áp dụng CSS (selector)
5. Một số tag HTML dùng riêng CSS

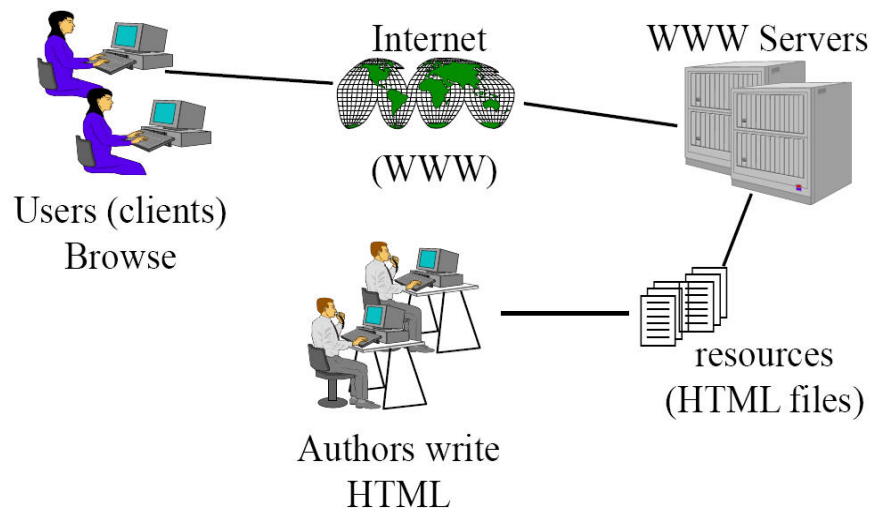
1 – Javascript cơ bản

2

Nội dung

1. DHTML
2. Ngôn ngữ client-script
3. Javascript cơ bản
4. Thực hành Javascript
5. .

Script Language & DHTML



Script Language

- **Script** : Ngôn ngữ lập trình thông dịch
Giúp website tương tác với người dùng
- **Client-side-script** : (Javascript, VBScript, ...)
 - Được tải về máy **client** và thực thi bằng **web browser**
 - Thiên về xử lý giao diện
- **Server-side-script** (Php, jsp, asp, aspx,...)
 - Được thực thi ở **web server** và trả kết quả về client
 - Thiên về xử lý truy cập hệ thống file trên server , kết nối với CSDL trên server.

Script Language

- Khi Browser gửi yêu cầu truy cập 1 trang server script
 - Trang server chứa **script server-side** và **client-side** (HTML, client-side script, css, ...)
 - Khi nhận được 1 yêu cầu, **web server (runtime-engine)** sẽ thực hiện các lệnh server script và gửi đến browser các trang web **DHTML** (chứa HTML, javascript, ...)

→ **Interactive & Dynamic web pages...**

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 (IEv6, Firefox 2.0, Opera 7.0, Netscape 4, ...)

Nhúng ngôn ngữ script vào webpage

- Định nghĩa Script trực tiếp

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

- Nhúng Script từ 1 file khác

```
<script src="xyz.js"></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 được thực thi khi trang web đang mở (sau khi thực thi các đoạn script trong phần `<head>`)
- Số lượng đoạn script không hạn chế.

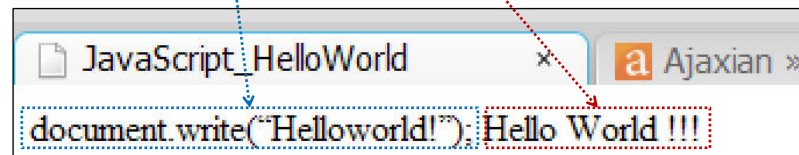
Javascript

- Giới thiệu Javascript
- Chức năng
- Cú pháp và quy ước
- Cấu trúc lập trình cơ bản
- Các ví dụ

Ví dụ kinh điển

Hello world

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



Javascript - Giới thiệu

- Web browser thực thi thẻ **script** khi load trang web HTML theo thứ tự từ trên xuống dưới
- Source code JavaScript có thể đặt trong file riêng (*.js) và nhúng (embed, include) vào file HTML bằng thuộc tính **scr** của thẻ **script**
- Code JavaScript được browser xử lý cùng thứ tự với các thẻ HTML. Trừ các **hàm (function)** chỉ được thực hiện khi có lời gọi hàm

Ví dụ

JavascriptIntro.html

```

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>JavaScript_Intro</title>
  <script>
    alert ("JavaScript in head!");
  </script>
</head>

<body>
  Before script. <br/>
  <script>
    alert("JavaScript in body!");
  </script>
  After script. <br/>
  Before external script. <br/>
  <script src="external.js">
    //Nothing here
  </script>
  After external script. <br/>
</body>

```

external.js

```

//JavaScript document
alert("JavaScript in external js file!");

```

The screenshots show the following sequence of events:

- JavaScript in head!**: Alert box triggered by the script in the head.
- JavaScript in body!**: Alert box triggered by the script in the body.
- JavaScript in external js file!**: Alert box triggered by the script in the external file.
- Before script. After script. Before external script. After external script.**: A sequence of four small alert boxes showing the execution flow.

1 – Javascript cơ bản13

Javascript - Chức năng

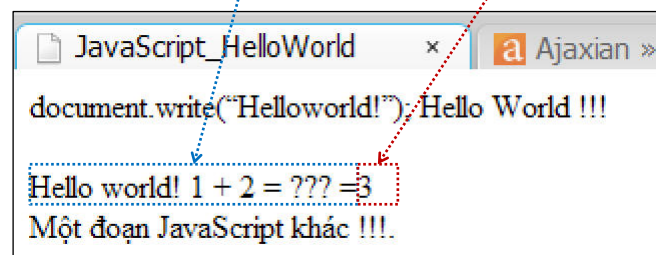
- Chức năng:
 - JavaScript: **client-script** mạnh mẽ và thông dụng
 - Lập trình và xử lý cơ bản cho 1 webpage
 - Kết hợp với **HTML DOM** để tương tác với người dùng (menu động, các nút đồ họa, các thay đổi màu sắc,...)
 - Kiểm tra dữ liệu nhập từ phía client (**Form Validation**)

Ví dụ

```
<script language="javascript" type="text/javascript">
    document.write("<br/>Hello world! 1 + 2 = ??? =" + {1 + 2});
</script>

<script>
    document.write("<br/>Một đoạn JavaScript khác !!!.");
</script>
```

Browser



1 – Javascript cơ bản

15

Javascript – Cú pháp và quy ước

1. Quy ước của ngôn ngữ lập trình
2. Biến và kiểu dữ liệu
3. Toán tử và phép so sánh
4. Cấu trúc điều khiển
5. Hàm
6. Một số đối tượng xây dựng sẵn (build-in)
7. Các hàm mở cửa sổ thông báo (pop-up windows)

1 – Javascript cơ bản

16

Javascript – Cú pháp và quy ước

Javascript có cú pháp, ký hiệu tương tự như Java và C++

- Phân biệt hoa thường
- Mọi lệnh phải kết thúc bằng dấu chấm phẩy ;
- Không phân biệt khoảng trắng, tab, xuống dòng trong câu lệnh.

//	Ghi chú 1 dòng	/* */	Ghi chú 1 khối
{ }	Khối lệnh	[]	Sử dụng cho mảng
()	Sử dụng cho hàm	\n, \t	Xuống hàng, ký tự Tab
\'	Ký tự nháy đơn trong chuỗi	\"	Ký tự nháy kép trong chuỗi

Chuỗi phân biệt trong dấu nháy đơn ' và dấu nháy kép "

Javascript – Phân biệt chuỗi với ' và "

```
<script>
    document.write("I said: 'Hello' # ");
    document.write('I said: "Hello" # ' + '<br>');
    document.write("I said: \"Hello' # ");
    document.write('I said: "Hello\' # ');
</script>
```

String In JavaScript

I said: 'Hello' # I said: "Hello" #
I said: "Hello' # I said: "Hello' #

```
<input size = "30" value='He said "Hello JavaScript"' />
<input type="button" value="Click me !" onclick="alert('Hello');"/>
```

He said hello

He said "Hello JavaScript" Click me !

Javascript – Biến và kiểu dữ liệu

```

<script>
  var num =3.4;
  document.write('Type of ' + num + ' is : ' + typeof(num) + '<br/>');
  var str="Anyone";
  document.write('Type of ' + str + ' is : ' + typeof(str) + '<br/>');
  var bool = true;
  document.write('Type of ' + bool + ' is : ' + typeof(bool) + '<br/>');
  var obj = new Array(3);
  obj[0] = 1; obj[1] = 2; obj[2] = 3;
  document.write('Type of ' + obj + ' is : ' + typeof(obj) + '<br/>');
</script>

```

```

<script type="text/javascript">
  var str="Visit Microsoft!";
  document.write(str.replace("Microsoft", "W3Schools"));
  document.write(str);
</script>

```

phương thức **KHÔNG** thay đổi giá trị

```

<script type="text/javascript">
  var fruits = ["Banana", "Orange", "Apple", "Mango"];
  document.write(fruits.reverse());
  document.write(fruits);
</script>

```

phương thức **ĐƯỢC** thay đổi giá trị

Parameters

Type of 3.4 is : number
 Type of Anyone is : string
 Type of true is : boolean
 Type of 1,2,3 is : object

1 – Javascript cơ bản
19

Javascript – Biến và kiểu dữ liệu

- Trong Javascript, mọi thứ là Object → từ khóa **new** để cấp phát. Một số kiểu dữ liệu cơ bản, không cần cấp phát. (Kiểu dữ liệu cơ bản)

Kiểu số	var y = 3.47; y.toPrecision(2);	y = new Number(1.22); y.toPrecision(5);
Kiểu chuỗi	var z = "Anyone"; z.length;	z = new String("dct"); z.length;
Kiểu bool	var b = false; if (b)	var b=new Boolean(false); if (b)

1 – Javascript cơ bản
20

Javascript – Biến và kiểu dữ liệu

- 1 Biến trong javascript có thể lưu bất kỳ kiểu dữ liệu nào. Biến có thể thay đổi kiểu dữ liệu trong quá trình hoạt động của biến

- Ví dụ:

```
var x = false ; // x kiểu boolean
x= "false" ; // x kiểu string
```

Javascript – Toán tử và phép so sánh

- Các toán tử số học: với $y = 3$

Operator	Description	Example	Result
+, -, *, /	Cộng, trừ, nhân, chia		
%	Modulus (div remainder)	$x = y\%2$	$x = 1$
++	Increment	$x = ++y$	$x = 4$
--	Decrement	$x = --y$	$x = 2$

- Toán tử ghép chuỗi **+** :

```
var x = 5, y = x + " tiết"; // y = "5 tiết"
var z = 2 + "+" + 3; // z = "2+3"
```

Javascript – Toán tử và phép so sánh

- Toán tử gán: $x = 10$ và $y = 5$

Operator	Example	Same as	Result (x)
=	$x = y$		5
+=	$x += y$	$x = x + y$	15
-=	$x -= y$	$x = x - y$	5
*=	$x *= y$	$x = x * y$	50
/=	$x /= y$	$x = x / y$	2
%=	$x \% = y$	$x = x \% y$	0

Javascript – Toán tử và phép so sánh

- Toán tử so sánh: $x = 5$

Operator	Description	Example
=	Bằng	$x = 8 \rightarrow \text{false}$
===	Chính xác bằng (value & type)	$x === "5" \rightarrow \text{false}$
!=	Khác	$x != 8 \rightarrow \text{true}$
>	Lớn hơn	$x > 8 \rightarrow \text{false}$
<	Nhỏ hơn	$x < 8 \rightarrow \text{true}$
>=	Lớn hơn hoặc bằng	$x >= 8 \rightarrow \text{false}$
<=	Nhỏ hơn hoặc bằng	$x <= 8 \rightarrow \text{true}$

Javascript - Cấu trúc điều khiển

- if và if {} else {}
- switch
- while
- do {} while
- for
- for .. in
- break và continue

Cấu trúc điều khiển – WHILE – DO WHILE

```
<script>
  var i = 5;

  while(i > 0)
  {
    document.write(i--);
  }
  document.write('<br>');

  do
  {
    document.write(--i);
  }
  while(i > 0);

  document.write('<br><hr>');
</script>
```

Cau Truc Dieu Khiem 1

54321

1

Phân biệt sự khác nhau giữa 2 cấu trúc

While (<ĐK>) {}

Kiểm tra ĐK đúng → thực hiện

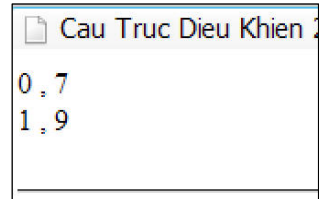
và

Do{} while (<ĐK>)

Thực hiện → kiểm tra ĐK đúng

Cấu trúc điều khiển – IF – IF ELSE

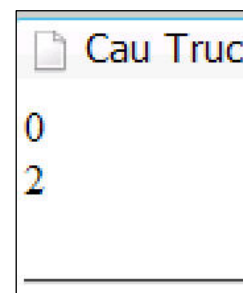
```
<script>
  var array = new Array();
  var i = -1;
  if (i % 2 == 0) // i là số chẵn
  {
    for(var j = 0; j < array.length; j++)
    {
      array[j] = i;
    }
  }
  else
  {
    array[0] = 7; array[1] = 9;
    for(i in array)
    {
      document.write(i + ' , ');
      document.write(array[i] + '<br>');
    }
  }
  document.write('<br><hr>');
</script>
```



Cấu trúc điều khiển - FOR

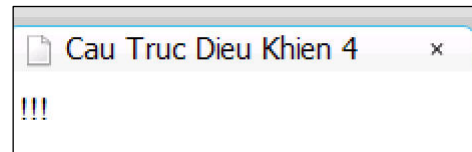
```
<script>
  var array = new Array();
  var i = -1;
  array.length = 5;
  for(i = 0; i < array.length; i++)
  {
    array[i] = i; // gán giá trị cho mảng
  }
  for(i = 0; i < array.length; i++)
  {
    if (i > 3)
      break;

    if (array[i] % 2 == 0)
      document.write(array[i] + '<br>');
    else
      continue;
  }
  document.write('<br><hr>');
</script>
```



Cấu trúc điều khiển – SWITCH CASE

```
<script>
  i= "c"; // switch can be applied to Number or String
  switch (i)
  {
    case "a":
      document.write('a');
      break;
    case "b":
      document.write('b');
      break;
    default:
      document.write('!!!');
  }
</script>
```



Javascript - Function

```
function tenHam(thamSo1,...)
{
  //Code
}
```

```
function tenHam(thamSo1,...)
{
  ...
  return (xxx);
}
```

- Hàm không cần khai báo kiểu trả về
- Hàm có thể trả về mọi kiểu giá trị
- Nên khai báo hàm trong thẻ head hoặc trong file **.js** để tái sử dụng
- Hàm chỉ được thực hiện khi có lệnh gọi
- Hàm thường kết với HTML DOM để xử lý sự kiện

Javascript - Function

```

<head>
<title>JavaScript Function</title>
<script>
function info(message)
{
    alert(message);
}

function sum(x, y)
{
    x = x + y;
    return x;
}
</script>
</head>
<body>
<script>
document.write('Calling info Function:');
info("FIT - SGU");
document.write('<br>Calling sum Function');

var x = 2, y = 3;
document.write('<br>x + y = ' + sum(x, y));
document.write('<br>x = ' + x);
</script>
</body>

```

Calling info Function:

[JavaScript Application]

FIT - SGU

OK

Calling info Function:
Calling sum Function
x + y = 5
x = 2

1 - Javascript cơ bản 31

Javascript - Một số đối tượng xây dựng sẵn

- String object
- Number object
- Date object
- Math object
- Array object
- Popup windows
- Một số hàm toàn cục (global functions) thường sử dụng

1 - Javascript cơ bản 32

Javascript - String			
var txt="Hello world!";			
Thuộc tính / Phương thức	Mô tả	Ví dụ / Giải thích	Giá trị
length	Chiều dài chuỗi	x = txt.length;	x==12
indexOf() lastIndexOf()	Tìm vị trí chuỗi con	x = txt.indexOf('o'); txt.lastIndexOf('o'); x = txt.indexOf('d'); x = txt.indexOf('D');	x==4 x==7 x==10 x== -1
substr() substring()	Lấy chuỗi con	txt=txt.substr(4); txt.substr(4, 2);	txt='o World!' txt = ???
replace()	Thay 1 chuỗi con bằng 1 chuỗi khác	txt=txt.replace('lo', ''); txt=txt.replace('o', 'a');	txt='Hel World!' txt='Hella World!'
1 – Javascript cơ bản			33

Javascript - String	
<ul style="list-style-type: none"> Lớp String của Javascript hỗ trợ 1 số phương thức để định dạng chuỗi tương tự như Javascript 	
<pre> <script> var txt="Hello world!"; document.write("<p>Bold: " + txt.bold() + "</p>"); document.write("<p>Bold: " + txt + "</p>"); </script> </pre>	
	
→ Không nên sử dụng các phương thức định dạng của lớp String	
1 – Javascript cơ bản	34

Javascript - Number

```
var num=45.678;
```

Thuộc tính / Phương thức	Mô tả	Ví dụ / Giải thích
toFixed()	Làm tròn số (quy định số chữ số của phần thập phân)	document.write(num.toFixed(2)); // 45.68
toPrecision()	Làm tròn số (quy định số chữ số)	document.write(num.toPrecision(3)); // 45.7
toExponential()	Đổi số thành dạng số khoa học	document.write(num.toExponential(2)); // 4.57e+1 var x = -15000; num = num.toPrecision(1); // -1.5e+4

Javascript - Math

Thuộc tính / Phương thức	Mô tả	Ví dụ / Giải thích
PI	Hằng số PI	document.write("PI: " + Math.PI.toFixed(2)); // PI: 3.14
E	Hằng số Euler	document.write("Euler: " + Math.E.toFixed(2)); // Euler: 2.72
ceil() floor() round()	Hàm làm tròn	x = Math.ceil(2.3); // x = 3 y = Math.floor(3.7); // x = 3 x = Math.round(-4.3); // x = ? y = Math.round(-4.7); // x = ?
pow()	Hàm lũy thừa	x = Math.pow(2, 3); x = 8
random()	Hàm phát sinh số ngẫu nhiên từ 0 đến 1	// phát sinh x ngẫu nhiên 1 → 10 x = Math.round(Math.random() * 9) + 1;
sin(), asin() cos(), acos() tan(), atan()	Các hàm lượng giác	...

Javascript - Array			
<pre>var arr = new Array(3); arr[0] = 2, arr[1] = 0, arr[2] = 1;</pre>			
Thuộc tính/ Phương thức	Mô tả	Ví dụ / Giải thích	Kết quả
length	Số phần tử của mảng	var x = arr.length; document.write(arr);	x == 3 2, 0, 1
sort()	Sắp xếp mảng	arr.sort(); document.write(arr); arr = arr.sort(); document.write(arr);	??? ???
slice()	Lấy 1 số phần tử của mảng	var arr1 = arr.slice(1); document.write(arr1); arr1 = arr.slice(0,1) document.write(arr1);	0, 1 0
1 – Javascript cơ bản			37

Javascript - Date		
<pre>var date = new Date();//date is the CURRENT date of the system</pre>		
Thuộc tính / Phương thức	Giá trị	Mô tả
getDate() getMonth() getFullYear() getHours() getMinutes() getSeconds()	1- 31 0 – 11 xxxx 0 – 23 0- 59 0 – 59	Lấy các thành phần của date
setFullYear() setDate() setMonth() setHours() setMinutes() setSeconds()	xxxx 1- 31 0 – 11 0 – 23 0- 59 0 – 59	Gán các thành phần của date Lưu ý: date.setFullYear(2009, 1,1);
<pre>var date1 = date; date.setDate(date.getDate() + 5); // date1 < date // tăng ngày lên 5 ngày, tự động đổi tháng, năm</pre>		
1 – Javascript cơ bản		38

Javascript – Hàm toàn cục (global functions)

Hàm	Mô tả
isNaN()	Kiểm tra 1 đối tượng có giá trị là số hay không
Number()	Đổi 1 đối tượng sang kiểu số
String()	Đổi 1 đối tượng sang kiểu chuỗi

```
var str = '3';
var x = 2.4;
document.write(str + x);
document.write('<br/>');
document.write(Number(str) + x);
document.write('<br/>');
document.write(str = String(x));
```

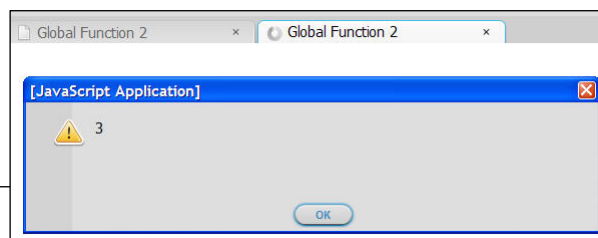
Global Functions

32.4
5.4
2.4

Javascript – Hàm toàn cục (global functions)

```
var str = '3';
var x = 2.4;

if(typeof(str) != 'number'){
    str = Number(str);
    alert(str);
}
document.write('<br/>');
document.write(str + x);
```



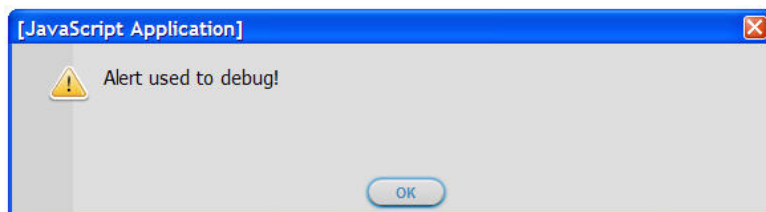
Global Function 2

5.4

Javascript – Các hàm mở cửa sổ (pop-up windows)

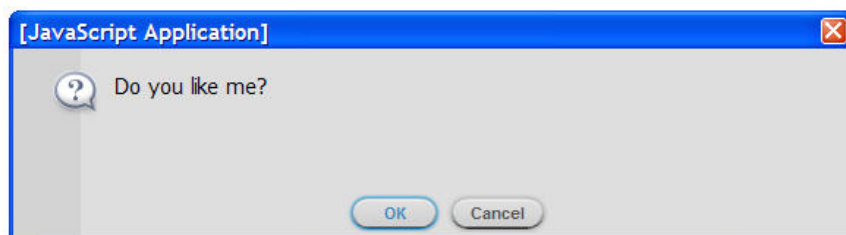
Hàm	Mô tả
alert(msg)	Mở cửa sổ thông báo
confirm(msg)	Mở cửa sổ chờ người dùng trả lời Yes / No
prompt(msg, default_value)	Mở cửa sổ cho người dùng nhập

```
alert('Alert used to debug!');
```



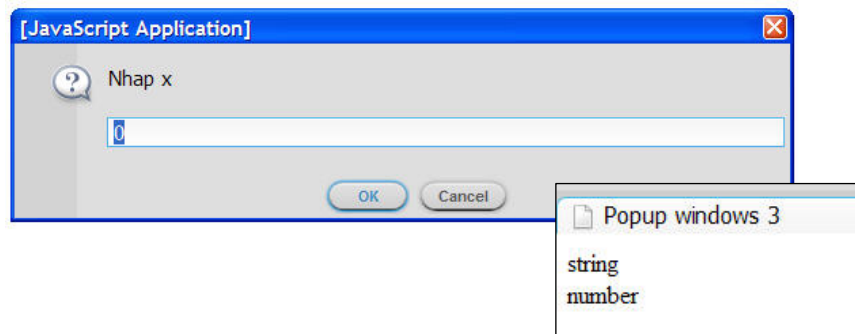
Javascript – Các hàm mở cửa sổ (pop-up windows)

```
var answer = confirm('Do you like me?');
alert(answer);
// ok      : answer = true
//cancel   : answer = false
```



Javascript – Các hàm mở cửa sổ (pop-up windows)

```
var x =prompt('Nhập x', 0);
while (isNaN(x))
    x = prompt('Nhập x',0);
document.write(typeof(x) + '<br/>');
//lúc này x có giá trị số nhưng là kiểu chuỗi
x = Number(x);
document.write(typeof(x));
```



Tham khảo

- <http://www.w3schools.com/js>
- <http://www.w3schools.com/jsref>
- <http://www.tizag.com/javascriptT/>
- <http://thuvienit.org>
- Google.... !!!



Bài thực hành

Đại Học Sài Gòn – Khoa CNTT

Javascript cơ bản

1 - Javascript cơ bản

GV: Phan Thị Kim Loan

Thực hành

- Demo code Javascript (Lưu ý cách xác định lỗi)
- Javascript không báo lỗi! Từ đoạn code có lỗi trở xuống sẽ KHÔNG được thực hiện
- → Debug bằng cách làm từng bước và **hàm alert**
- Bài tập code Javascript: [BTJavascript.pdf](#)

1 – Javascript cơ bản

46

