

<b>CHƯƠNG 1 TỔNG QUAN</b>	1
1.1 GIỚI THIỆU	1
1.2 CÁCH ĐẶT MÃ SCRIPT TRONG TRANG WEB	1
Ví dụ 1.1	1
Ví dụ 1.2	2
<b>CHƯƠNG 2 NGÔN NGỮ JAVASCRIPT</b>	3
2.1 KIỂU DỮ LIỆU	3
2.2 KHAI BÁO BIẾN	3
2.3 TOÁN TỬ	3
2.4 CHÚ THÍCH	4
2.5 CHÈN THẺ HTML VÀ KÝ TỰ ĐẶC BIỆT	4
Ví dụ 2.1	4
<b>CHƯƠNG 3 CÁC LỆNH ĐIỀU KHIỂN</b>	5
3.1 LỆNH Rẽ NHÁNH	5
3.1.1 Lệnh if	5
3.1.2 Lệnh switch ... case	5
3.2 LỆNH LẶP	5
3.2.1 Lệnh while	5
3.2.2 Lệnh do...while	6
3.2.3 Lệnh for	6
Ví dụ 3_1	6
<b>CHƯƠNG 4 HÀM</b>	7
4.1 XÂY DỰNG HÀM	7
4.2 MỘT SỐ HÀM CỦA JAVASCRIPT	7
Ví dụ 4_1	7
Ví dụ 4_2	8
<b>CHƯƠNG 5 ĐỐI TƯỢNG</b>	9
5.1 GIỚI THIỆU	9
5.2 CÁC ĐỐI TƯỢNG TRONG TRANG WEB	9
5.2.1 Đối tượng window	9
Ví dụ 5.1	11
Ví dụ 5.2	12
5.2.2 Đối tượng document	13
5.3 CÁC ĐỐI TƯỢNG JAVASCRIPT	14
5.3.1 Đối tượng String	14
Ví dụ 5.3	15
5.3.2 Đối tượng Math	16
Ví dụ 5.4	17
5.3.3 Đối tượng Date	17
Ví dụ 5.5	18
5.3.4 Đối tượng mảng	18
Ví dụ 5.6	19
5.3.5 Đối tượng form	19
Ví dụ 5.7	20
5.4 XÂY DỰNG ĐỐI TƯỢNG MỚI	23
Ví dụ 5.8	23

## CHƯƠNG 1 TỔNG QUAN

### 1.1 GIỚI THIỆU

Javascript không phải là Java, Javascript là một ngôn ngữ kịch bản (script) phía server (server side) và phía client (client side) (thông dụng hơn).

Ngôn ngữ kịch bản là ngôn ngữ lập trình kiểu thông dịch. Các đoạn chương trình viết bằng ngôn ngữ này được nhúng trong các trang web sẽ được các trình duyệt thông dịch để thực hiện. Các ngôn ngữ kịch bản cho phép phát triển các chương trình đơn giản nhanh và dễ dàng hơn là các ngôn ngữ lập trình dạng biên dịch C, C++... Tuy nhiên lại khó phát triển các ứng dụng lớn, phức tạp.

Javascript là ngôn ngữ hướng đối tượng. Có cú pháp gần giống C++. Phân biệt chữ hoa, chữ thường. Có thể viết câu lệnh trên nhiều dòng hoặc nhiều câu lệnh trên một dòng nhưng phải ngăn cách các câu lệnh bởi dấu chấm phẩy. Lưu ý: không bắt buộc phải có dấu chấm phẩy cuối mỗi câu lệnh.

Javascript hoạt động tốt trên hầu hết các trình duyệt web.

Có thể dùng chương trình soạn thảo: Notepad, Frontpage, Visual InterDev ... để viết các đoạn chương trình Javascript.

### 1.2 CÁCH ĐẶT MÃ SCRIPT TRONG TRANG WEB

**Cách 1: Chèn trực tiếp đoạn mã lệnh Javascript vào trang web.**

Đặt các dòng mã lệnh của Javascript giữa cặp thẻ <script>...</script> vào bất kỳ vị trí nào trong trang web.

#### Ví dụ 1.1

```
<html>
<head>
    <script language="javascript">
        <!--
        document.write("Chao ban a");
        -->
    </script>
</head>
<body>
    <script language="javascript">
        <!--
        document.write("Chao ban b");
        -->
    </script>
</body>
```

**Cách 2: Có thể viết một tập tin Javascript riêng và sau đó kết nối với một hoặc nhiều tập tin trang web khác nhau.**

[Ví dụ 1.2](#)

Bước 1:

Soạn thảo tập tin chứa đoạn chương trình Javascript như sau:

```
document.write("Chao ban a");
```

Lưu lại với tên vd1\_2.js

Bước 2:

Soạn thảo trang web liên kết với tập tin trên

```
<html>
```

```
<body>
```

```
<script language="javascript" src="vd1_2.js">
```

```
</script>
```

```
</body>
```

```
</html>
```

## CHƯƠNG 2 NGÔN NGỮ JAVASCRIPT

### 2.1 KIỂU DỮ LIỆU

Các kiểu dữ liệu cơ bản: kiểu chuỗi, kiểu số, kiểu luận lí, ...

### 2.2 KHAI BÁO BIẾN

Dùng từ khóa var để khai báo một biến. Khi khai báo biến trong Javascript không cần xác định kiểu dữ liệu cho biến do đó khi một biến được khai báo xong nó có thể chứa bất kỳ kiểu dữ liệu nào.

Cách đặt tên cho biến:

- Tên biến phải bắt đầu bằng ký tự hoặc gạch dưới (\_)
- Không được bắt đầu bằng ký tự số.
- Không chứa khoảng trống.
- Không trùng với bộ từ khóa của Javascript.

Ví dụ

Khai báo nhiều biến: `var x,y;`

Khai báo và gán giá trị, x là một biến kiểu số nguyên: `var x=5;`

Gán lại x mang giá trị của một chuỗi: `x= "Hello";`

Khai báo mảng

```
var a=new Array(10) //một chiều có 10 phần tử chỉ số bắt đầu là 0
```

```
var a=new Array()
```

```
var a=new Array("t0","t1","t2","t3")
```

Tầm vực của biến:

- Biến toàn cục: tầm ảnh hưởng của biến là trong toàn bộ chương trình, nó được khai báo ngoài các hàm.
- Biến cục bộ: được khai báo trong hàm, tầm ảnh hưởng trong hàm mà biến này được khai báo.

### 2.3 TOÁN TỬ

Toán tử số học: `+, -, *, /, %, ++, --`

Toán tử gán: `=, +=, -=, *=, /=, %=`

Toán tử so sánh: `==, !=, >, <, >=, <=`

Toán tử logic: `&&, ||, !`

Toán tử chuỗi: `+`

Ví dụ:

```
t1="Xin chào"
```

```
t2="An"
```

```
t3=t1+" "+t2
```

Biến t3 sẽ chứa giá trị là: "Xin chào An".

Toán tử điều kiện

(Điều kiện) ? giá trị 1: giá trị 2

Ví dụ:

```
Solon=(a>b)?a:b
```

## 2.4 CHÚ THÍCH

Có thể thêm dấu chú thích để chú thích một dòng lệnh // hoặc nhiều dòng /\*...\*/

## 2.5 CHÈN THẺ HTML VÀ KÝ TỰ ĐẶC BIỆT

### Ví dụ 2.1

```
<body>
```

```
<pre>
```

```
<script language="javascript">
```

```
    document.write("<img src=../image/HoaNho.gif>")
```

```
    document.write("<br><b><i>Chào bạn An</i></b>")
```

```
    document.write("\n\tChào bạn An")
```

```
    document.write("\nChào bạn \"An\"")
```

```
    document.write("\nThẻ \<br> xuống dòng")
```

```
</script>
```

```
</pre>
```

```
</body>
```

## CHƯƠNG 3 CÁC LỆNH ĐIỀU KHIỂN

### 3.1 LỆNH RỄ NHÁNH

#### 3.1.1 Lệnh if

Cú pháp

```
if (điều kiện)
{
    //thực hiện mã lệnh ở đây nếu điều kiện đúng
}
else
{
    //thực hiện mã lệnh ở đây nếu điều kiện sai
}
```

#### 3.1.2 Lệnh switch ... case

Cú pháp

```
switch (biểu thức)
{
    case <giá trị 1>:
        //thực hiện mã lệnh ở đây nếu giá trị biểu thức=giá trị 1
        break;
    case <giá trị 2>:
        //thực hiện mã lệnh ở đây nếu giá trị biểu thức=giá trị 2
        break;
    default:
        //thực hiện mã lệnh ở đây nếu giá trị biểu thức không phải là giá trị1 hay
        //giá trị 2
        break;
}
```

### 3.2 LỆNH LẶP

#### 3.2.1 Lệnh while

Cú pháp

```
while (điều kiện)
{
    //thực hiện mã lệnh ở đây nếu giá trị điều kiện đúng
}
```

### 3.2.2 Lệnh do...while

Cú pháp

```
do
{
    //thực hiện mã lệnh ở đây sau đó kiểm tra điều kiện nếu đúng thì thực
    //hiện lại mã lệnh này
}
while (điều kiện)
```

### 3.2.3 Lệnh for

Cú pháp

```
for (giá trị khởi đầu của biến điều khiển ; điều kiện ; giá trị tăng hoặc giảm biến điều khiển)
{
    //đoạn mã lệnh
}
```

#### Ví dụ 3.1

```
<script language="javascript">
var i,j;
document.write("<table border=1>");
for(i=1;i<=2;i++)
{
    document.write("<tr>");
    for(j=1;j<=3;j++)
        document.write("<td>dòng "+i+" cột "+j+"</td>");
    document.write("</tr>");
}
document.write("</table>")
</script>
```

## CHƯƠNG 4 HÀM

### 4.1 XÂY DỰNG HÀM

Cú pháp

```
function <tên hàm>(tham số 1, tham số 2,...)
{
    //đoạn mã lệnh
}
```

Ví dụ

```
function cong(a,b)
{
    c=a+b
    return c
}
```

Gọi hàm: tongcong=cong(2,3)

### 4.2 MỘT SỐ HÀM CỦA JAVASCRIPT

eval(s): trả về giá trị số của s.

isNaN(s): trả về true nếu s không phải là một số, false trong trường hợp ngược lại.

parseInt(s): trả về một số nguyên từ s. Nếu như s theo sau là các ký tự chữ thì các ký tự này sẽ bị bỏ qua. Nếu s bắt đầu bằng ký tự chữ thì hàm trả về giá trị NaN

parseFloat(s): trả về một số thực từ s. Nếu như s theo sau là các ký tự chữ thì các ký tự này sẽ bị bỏ qua. Nếu s bắt đầu bằng ký tự chữ thì hàm trả về giá trị NaN.

#### Ví dụ 4.1

```
var s1= "123",s2= "456",kq;
kq=s1+s2;
document.writeln(kq);
kq=eval(s1)+eval(s2)
document.writeln(kq);
```

```
var s= "123.4",kq;
kq=parseInt(s)
document.writeln(kq);
s= "12a3.4"
kq=parseInt(s)
document.writeln(kq);
```



```
s="a123.4"
kq=parseInt(s)
document.writeln(kq);
var s= "123.4",kq;
kq=parseFloat(s);
document.writeln(kq);
```

#### Ví dụ 4\_2

```
<script language="javascript">
var n;
do
{
n=prompt("Nhập vào một số bất kỳ","");
if(isNaN(n))
    document.write("Bạn phải nhập số, xin nhập lại");
else
{
    document.write("Bạn đã nhập số "+n);
    break;}
}
while(1)
</script>
```

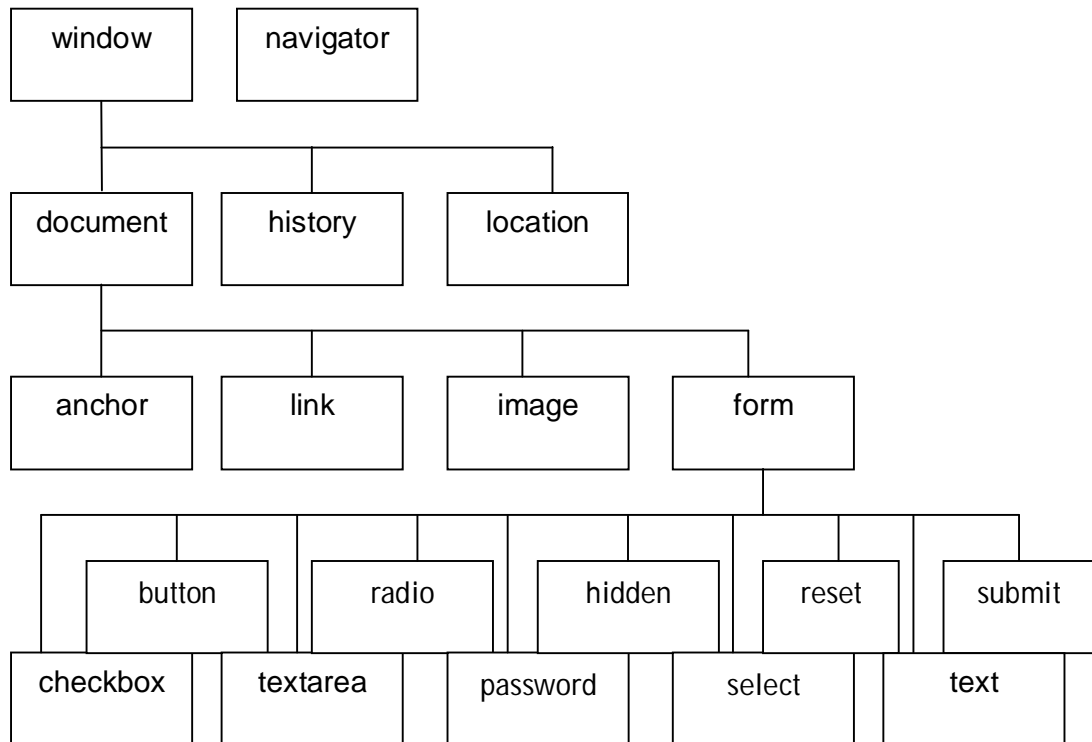
## CHƯƠNG 5 ĐỐI TƯỢNG

### 5.1 GIỚI THIỆU

Trình duyệt tổ chức tất cả các đối tượng theo tổ chức phân cấp hình cây. Javascript có thể thao tác lên các đối tượng này.

Ngoài một số đối tượng do chương trình xây dựng sẵn, chúng ta có thể tự tạo cho mình những đối tượng cần thiết.

Một thẻ HTML cũng được xem như là một đối tượng và các thuộc tính của thẻ đó được xem như là đối tượng con của nó.



Tài liệu này chỉ giới thiệu một số đối tượng cùng với một số thuộc tính và phương thức thông dụng của nó.

### 5.2 CÁC ĐỐI TƯỢNG TRONG TRANG WEB

#### 5.2.1 Đối tượng window

Thuộc tính	Ý nghĩa
status	Trả về hoặc gán chuỗi ký tự hiển thị trên thanh trạng thái cuối cửa sổ.
defaultStatus	Gán chuỗi ký tự mặc định hiển thị trên thanh trạng thái ở cuối cửa sổ. Vd:

	<code>window.defaultStatus="Chào mừng bạn truy cập web site"</code>
Parent	Tham chiếu tới trang đang gọi frame hiện hành được tạo bởi thẻ <code>&lt;Frameset&gt;</code> . Dùng parent để truy xuất tới các frame khác được tạo cùng thẻ <code>&lt;Frameset&gt;</code> . Đây là thuộc tính của frame và window.
self or top	Tham chiếu đến cửa sổ hiện hành.

Phương thức	Ý nghĩa
<code>alert(s)</code>	Hiển thị hộp thoại gồm chuỗi và nút OK
<code>confirm(s)</code>	Hiển thị hộp thoại gồm chuỗi và hai nút OK + Cancel. Hàm trả về giá trị là true khi người nhấn OK và false khi nhấn Cancel.
<code>prompt(s,mặcdịnh)</code>	Hiển thị hộp thoại có một hộp text chờ nhận dữ liệu. Hàm trả về giá trị mà người sử dụng nhập vào. Có thể gán chuỗi mặc định trong hộp text.
<code>open(url,tên,thuộctính)</code>	Mở một cửa sổ mới với tên lưu trong đối số tên và nạp tài liệu trong url vào cửa sổ, có thể thay đổi một số thuộc tính của cửa sổ,
<code>close()</code>	đóng cửa sổ Ví dụ: <code>close()</code> , <code>self.close()</code> : đóng cửa sổ hiện hành. <code>a.close()</code> : đóng cửa sổ tên a.
<code>focus()</code>	Đưa con trỏ về cửa sổ
<code>blur()</code>	Di chuyển con trỏ khỏi cửa sổ
<code>clearTimeout(id)</code>	Hủy thời gian đã thiết lập bằng <code>setTimeout()</code>
<code>setTimeout(lệnh,thời gian)</code>	Yêu cầu thực hiện lệnh sau một khoảng thời gian tính bằng mili giây. Hàm trả về giá trị id (duy nhất đối với mỗi hàm <code>setTimeout()</code> thực hiện một lệnh). Giá trị id này dùng để xóa khoảng thời gian đã thiết lập nếu không cần thực hiện hàm <code>setTimeout</code> nữa.
<code>scroll(x,y)</code>	di chuyển cửa sổ đến toạ độ x,y

Sự kiện	Ý nghĩa
<code>onBlur</code>	xảy ra khi một thành phần trên form mất con trỏ
<code>onChange</code>	xảy ra khi có sự thay đổi nội dung một thành phần trên form trước khi nó bị mất con trỏ

onClick	xảy ra khi nhấp mouse trên một đối tượng
onFocus	xảy ra khi một thành phần form nhận con trỏ
onMouseOver	xảy ra khi con trỏ chuột được đặt trên một đối tượng liên kết
onSelect	được kích hoạt bằng cách chọn một số hay tất cả văn bản trong text hay textarea
onSubmit	được kích hoạt khi người dùng đệ trình form
onLoad	xảy ra khi trình duyệt hoàn thành việc nạp một cửa sổ hay tất cả các frame trong thẻ <FRAMESET>. Đây là biến cố xử lý của window. Biến cố thường được đặt trong thẻ <BODY> hoặc <FRAMESET>
onUnload	xảy ra khi thoát một trang. Biến cố thường được đặt trong thẻ <BODY> hoặc <FRAMESET>

### Ví dụ 5.1

```

<body>
<HR>
<FORM>
<INPUT TYPE="button" VALUE="mở cửa sổ mới"
onClick="vd1_1=window.open('vd1_1.htm');">
<P><INPUT TYPE="button" VALUE="đóng cửa sổ mới mở"
onClick="vd1_1.close();" >
<P><INPUT TYPE="button" VALUE="đóng cửa sổ hiện tại"
onClick="window.close();">
<HR>
<P><INPUT TYPE="button"
VALUE="mở cửa sổ có định dạng"
onClick="Win=window.open('vd1_1.htm','Win',
'menubar=no,toolbar=no,location=no,scrollbars=no,status=no,
width=400,height=200,resizable=yes');">
</FORM>
<HR>
</body>

```

Ví dụ 5.2

```
<html>
<head>
<script>
var idq;
function thuchien()
{
var arrhinh= new Array();
arrhinh[0]= "../image/hoanho.gif";
arrhinh[1]= "../image/hinhnguoi.gif";
arrhinh[2]= "../image/birthday.gif";
var i=Math.round(Math.random()*3)
if(i==3)
    i=0;
hinh.src=arrhinh[i];
idq=setTimeout("thuchien()",1000);
}
function dung()
{
clearTimeout(idq);
}
</script>
</head>
<body>

<form>
<input type=button value="thực hiện" onClick="thuchien()">
<input type=button value="dừng" onClick="dung()">
</form>
</body>
</html>
```

### 5.2.2 Đối tượng document

Thuộc tính	Ý nghĩa
Title	Trả về giá trị tiêu đề trang
bgColor	Thiết lập hoặc trả về giá trị màu nền trang
fgColor	Thiết lập hoặc trả về giá trị màu văn bản Ví dụ: document.fgColor="red";
alinkColor	Thiết lập hoặc trả về giá trị màu của liên kết đang xem của trang
vlinkColor	Thiết lập hoặc trả về giá trị màu của liên kết đã xem của trang
linkColor	Thiết lập hoặc trả về giá trị màu của liên kết
Location	URL của trang hiện hành
lastModified	Trả về giá trị ngày giờ cuối cùng mà tài liệu được cập nhật

Phương thức	Ý nghĩa
Write	Xuất chuỗi ra trang hiện hành.
WriteLn	Xuất chuỗi ra trang hiện hành sau đó xuống dòng mới.

## 5.3 CÁC ĐỐI TƯỢNG JAVASCRIPT

### 5.3.1 Đối tượng String

Phương thức	Ý nghĩa
<b>bold()</b>	Trả về một chuỗi in đậm Ví dụ: str.bold()
<b>italics()</b>	Trả về một chuỗi in nghiêng
<b>fontcolor()</b>	Trả về một chuỗi với màu đã được xác lập.
<b>CharAt(d)</b>	Trả về ký tự thứ d trong chuỗi, d từ 0 đến str.length-1
<b>indexOf(s,[d])</b>	Trả về vị trí đầu tiên được tìm thấy của chuỗi s bắt đầu tìm từ vị trí d. Nếu không tìm thấy thì hàm trả về giá trị -1
<b>lastIndexOf(s)</b>	Trả về vị trí của cuối cùng được tìm thấy của chuỗi s bắt đầu tìm từ phải qua trái. Nếu không tìm thấy thì hàm trả về giá trị -1
<b>substring(d1, d2)</b>	Tách ra một chuỗi con từ một chuỗi. Nếu d1<d2 chuỗi trả về từ vị trí d1 đến d2 - 1 Nếu d1>d2 chuỗi trả về từ vị trí d2 đến d1 - 1 Nếu d1=d2 chuỗi trả về là null.
<b>big()</b>	Trả về một chuỗi đặt trong cặp thẻ <big>
<b>small()</b>	Trả về một chuỗi nhỏ hơn
<b>strike()</b>	Trả về một chuỗi được gạch ngang qua thân chuỗi
<b>sub()</b>	Trả về một chuỗi kiểu subscript
<b>sup()</b>	Trả về chuỗi kiểu superscript
<b>toLowerCase()</b>	Chuyển chuỗi thành chữ thường
<b>toUpperCase()</b>	Chuyển chuỗi thành chữ hoa
<b>match(s)</b>	Tương tự như hàm indexOf và lastIndexOf, nhưng phương thức này trả về một chuỗi cụ thể nếu không tìm thấy thì trả về giá trị "null".
<b>replace(s1,s2)</b>	Thay thế một vài ký tự bằng một vào ký tự mới
<b>search(s)</b>	Trả về vị trí chuỗi được tìm thấy, nếu không tìm thấy thì trả về giá trị -1
<b>slice(d)</b>	Trả về một chuỗi con được cắt từ chuỗi mẹ tại vị trí cắt

substr(d1,d2)	Trả về chuỗi con bắt đầu từ vị trí d1 và có chiều dài d2.
---------------	---

Thuộc tính	Ý nghĩa
Length	Chiều dài chuỗi

### Ví dụ 5.3

```

<script language="JavaScript">
document.writeln("Học JavaScript ".bold());
document.writeln("Học JavaScript ".italics());
document.writeln("Học JavaScript ".fontcolor("red"));
document.writeln("Học JavaScript ".charAt(0));
document.writeln("Học JavaScript ".charAt(5));
document.writeln("Học JavaScript ".lastIndexOf("a"));
document.writeln("Học JavaScript ".indexOf('A'));
document.writeln("Học JavaScript ".indexOf('a'));
document.writeln("Học JavaScript ".indexOf('a',6));
document.writeln("Học JavaScript ".indexOf('av'));
document.writeln("Học JavaScript ".indexOf('aa'));
document.writeln("Học JavaScript ".indexOf('a',11));
document.writeln("Học JavaScript ".substring(5,0));
document.writeln("Học JavaScript ".big());
document.writeln("Học JavaScript ".small());
document.writeln("Học JavaScript ".strike());
document.writeln("Học JavaScript "+"1.2".sub());
document.writeln("Học JavaScript "+"1.2".sup());
document.writeln("Học JavaScript ".toLowerCase());
document.writeln("Học JavaScript ".toUpperCase());
document.writeln("Học JavaScript ".match("a"));
document.writeln("Học JavaScript ".match("Java"));
document.writeln("Học JavaScript ".match("java"));
document.writeln("Học JavaScript ".replace("a","b"));
document.writeln("Học JavaScript ".search("a"));
document.writeln("Học JavaScript ".slice(5));

```



```
document.writeln("Học JavaScript".substr(5,7));
document.writeln("Học JavaScript".length);
document.writeln("Học JavaScript".substring(5,7).length);
</script>
```

### 5.3.2 Đối tượng Math

Thuộc tính	Ví dụ
PI	Math.PI
E	Math.E
LN2	Math.LN2

Phương thức	Mô tả
sqrt(x)	Trả về giá trị căn bậc 2 của x
abs(x)	Trả về giá trị tuyệt đối của biến x
round(x)	Làm tròn số x
ceil(x)	Trả về số nguyên lớn hơn hoặc bằng x
floor(x)	Trả về số nguyên nhỏ hơn hoặc bằng x
log(x)	Trả về giá trị log của x
max(x,y)	Trả về giá trị lớn nhất trong hai số x và y
min(x,y)	Trả về giá trị nhỏ nhất trong hai số x và y
pow(x,y)	Trả về giá trị x lũy thừa y
random()	Trả về giá trị một số ngẫu nhiên từ 0 đến 1
log(x)	Trả về giá trị log của x
sin(x)	Trả về giá trị sine của x
cos(x)	Trả về giá trị cosine của x
asin(x)	Trả về giá trị asine của x
acos(x)	Trả về giá trị arccosine của x
atan(x)	Trả về giá trị arctangent của x

### Ví dụ 5.4

```
<script language="JavaScript">
document.writeln(Math.sqrt(9));
document.writeln(Math.abs(-79));
document.writeln(Math.round(7.567));
document.writeln(Math.round(7.467));
document.writeln(Math.ceil(7.12));
document.writeln(Math.floor(7.95));
document.writeln(Math.max(7,9));
document.writeln(Math.min(7,9));
document.writeln(Math.pow(6,2));
document.writeln(Math.random());
document.writeln(Math.sin(90));
document.writeln(Math.cos(90));
document.writeln(Math.PI);
document.writeln(Math.E);
document.writeln(Math.LN2);
```

### 5.3.3 Đối tượng Date

Ví dụ: Khởi tạo đối tượng ngày

```
var ngay1, ngay2, n;
ngay1 = new Date("March 25, 2004 7:45:5");
ngay2 = new Date(2004,3,25,7,45,5);
n = new Date();
```

Phương thức	Ý nghĩa
Date()	Khởi tạo đối tượng ngày
getDate()	ngày của tháng (1-31)
getDay()	ngày của tuần (0-6) Sunday=0)
getMonth()	tháng (0–11) (January=0)
getFullYear()	Năm
getFullYear()	Năm

getHours()	giờ của ngày (0–23)
getMinutes()	phút của giờ (0–59)
getSeconds()	Giây (0–59)
getMilliseconds()	Mili giây (0–999)
setDate(n) set....	đặt ngày của tháng là n
toString()	đổi một đối tượng ngày thành chuỗi
toGMTString()	đổi một đối tượng ngày thành chuỗi theo múi giờ quốc tế.

Ví dụ 5.5

```
document.writeln(ngay.getDay());
document.writeln(ngay.getDate());
document.writeln(ngay.getMonth());
document.writeln(ngay.getYear());
document.writeln(ngay.getHours());
document.writeln(ngay.getMinutes());
document.writeln(ngay.getSeconds());
document.writeln(ngay.toString());
document.writeln(ngay.toGMTString());
```

**5.3.4 Đối tượng mảng**

Thuộc tính	Ý nghĩa
Length	Trả về số phần tử của mảng
Phương thức	Ý nghĩa
sort()	Sắp xếp các phần tử trong mảng
reverse()	Đảo ngược thứ tự các phần tử của mảng.
pop()	Lấy đi phần tử cuối cùng của mảng
push("a","b")	Thêm một hoặc nhiều phần tử vào cuối mảng
concat()	Dùng để ghép nối hai hoặc nhiều mảng lại với nhau

join(separator)	Ghép nối các phần tử trong mảng lại với nhau ngăn cách nhau bởi dấu separator
-----------------	---

**Ví dụ 5.6**

```

var ds = new Array(3)
ds[0] = "Lê Văn"
ds[1] = "Nam"
ds[2] = "An"
for (i=0; i<ds.length; i++)
{
    document.write(ds[i] + "<br>")
}
document.write(ds.reverse() + "<br>")

```

**5.3.5 Đối tượng form**

Thành phần	Sự kiện
Button	onClick
Checkbox	onClick
Form	OnSubmit, onReset
Textbox	OnBlur, OnChange, OnFocus, Onselect
Radio	OnClick
Reset button	OnClick
Dropdown menu	OnBlur, onChange, onFocus, onSelect
Submit button	OnClick
Textarea	OnBlur, OnChange, OnFocus, Onselect

Cú pháp truy cập giá trị các phần tử trên form:

```

document.tênform.tênthànhphầncủatform.thuộctính
document.tênform.tênthànhphầncủatform.phươngthức

```

Thao tác trên mảng các thành phần của form

Thuộc tính	Mô tả
Value	Xác định giá trị của phần tử thứ j trong form i. Ví dụ: document.tênform[i].tênthànhphầncủaform[j].value
Checked	Xác định phần tử thứ j có được checked không. Nếu có trả về giá trị true còn không trả về giá trị false Ví dụ: document.tênform[i].tênthànhphầncủaform[j].checked
Disable	Thiết lập chế độ mờ: gán giá trị true không cho phép người sử dụng chọn lựa và ngược lại. Ví dụ: document.tênform.tênthànhphầncủaform.disable=true
isDisable	Kiểm tra phần tử có mờ không ( true là mờ và ngược lại) Ví dụ: if(document.tênform.tênthànhphầncủaform.isDisable==true)

Thao tác trên dropdownmenu

Thuộc tính	Ý nghĩa
length	Trả về số phần tử trong danh sách dropdownmenu.
selectedIndex su dụng cho list box	trả về chỉ số của phần tử được chọn trong danh sách

Phương thức	Ý nghĩa
Focus	Đưa con trỏ về lại text box hoặc dropdownmenu document.tênform.tênthànhphầncủaform.focus()

[Ví dụ 5.7](#)

**Mời nhập thông tin**

Địa chỉ Email

Phái ☐ Nữ ☐ Nam

Nghề nghiệp

```
<body>
<script language="JavaScript">
<!--
var ketqua;
var phai=new Array("Nữ","Nam");
function TaoRadio(ten,mang) {
    var s="";
    for(var i=0; i<mang.length; i++) {
        s+= "<INPUT NAME=\" " + ten + " \" + \"TYPE=radio VALUE=\" "+
mang[i] + "\">" + mang[i];    }
    return s;
}
function ktDiaChi() {
    if(document.form1.diachi.value == ""){
        alert("\nBạn chưa nhập địa chỉ email.\n");    return false;
    }
    if(document.form1.diachi.value.indexOf("@") == -1){
        alert("\nBạn nhập địa chỉ email không hợp lệ.\n");    return false;
    }
    ketqua+= " Địa chỉ email: "+document.form1.diachi.value;
    return true;
}
function ktPhai() {
    var j = -1;
    for(var i=0; i<document.form1.p.length; i++)
        if(document.form1.p[i].checked) {
            j = i;    break;
        }
    if(j == -1) {
        alert("\nBạn chưa chọn phái.\n");    return false;
    }
    ketqua+="\nBạn là phái "+document.form1.p[j].value;
    return true;
```

```
}  
function kiemtra() {  
    ketqua="";  
    if(!ktDiaChi())      return false;  
    if(!ktPhai())        return false;  
    alert(ketqua);  
    return true;  
}  
-->  
</script>  
<FORM method=post name=form1>  
<CENTER>  
<H3>Mời nhập thông tin</H3>  
<TABLE border=0 width="50%" >  
  <TR>  
    <TD vAlign=top>Địa chỉ Email</TD>  
    <TD>  
      <SCRIPT language=JavaScript>  
        <!--  
          document.write("<input name=diachi>" + "<BR>");  
        -->  
      </SCRIPT>  </TD> </TR>  
  <TR>  
    <TD vAlign=top>Phái</TD>  
    <TD>  
      <SCRIPT language=JavaScript>  
        <!--  
          document.write(TaoRadio("p",phai) + "<BR>");  
        -->  
      </SCRIPT>  </TD>  </TR>  
</TABLE><BR>  
<INPUT type=button value="Đăng ký" align=left onclick=kiemtra()>  
</FORM></CENTER> </body>
```

## 5.4 XÂY DỰNG ĐỐI TƯỢNG MỚI

Ngoài các đối tượng đã được xây dựng sẵn trong Javascript, người lập trình có thể tự tạo các đối tượng mới với các phương thức và thuộc tính riêng cho đối tượng đó.

Bước 1: Định nghĩa đối tượng bằng cách xây dựng hàm.

Bước 2: Tạo các thực thể (instance) cho đối tượng bằng cách dùng từ khóa new.

### Ví dụ 5.8

```
<script language="javascript">
function Xuat()
{
    document.write("<br>Tên: "+this.ten);
    document.write("<br>Tuổi: "+this.tuoi);
}
function SinhVien(t1,t2)
{
    this.ten=t1;
    this.tuoi=t2;
    this.Xuat=Xuat;
}
var n=new SinhVien("An",18);
var m=new SinhVien();
m.ten="Bình";
m.tuoi=20;
n.Xuat() ;
m.Xuat();
</script>
```