

CHƯƠNG VIII

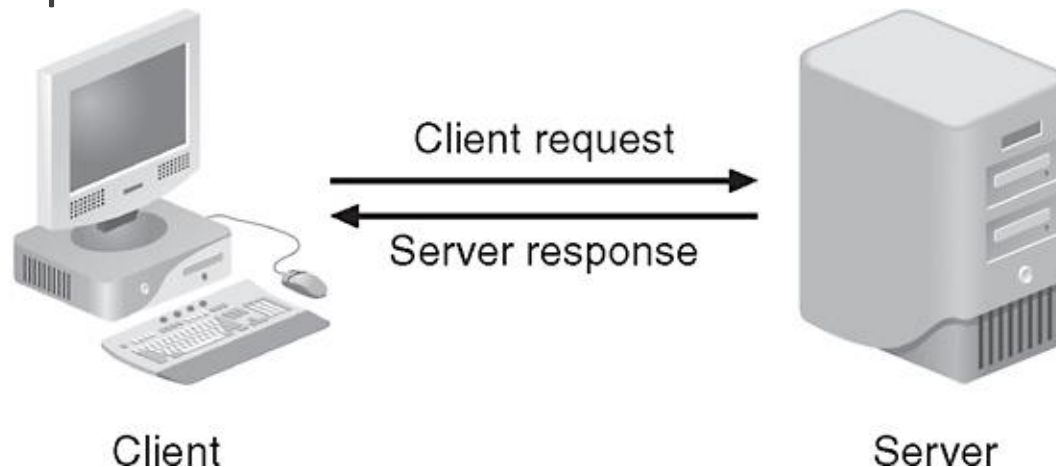
JAVASCRIPT

Nội dung

- ❑ Giới thiệu về Javascript
- ❑ Biến và dữ liệu
- ❑ Toán tử và biểu thức
- ❑ Các lệnh trong Javascript
- ❑ Hàm trong Javascript
- ❑ Cấu trúc điều khiển

Giới thiệu

- ❑ Javascript ra đời với tên gọi LiveScript, sau đó Netscape đổi tên thành Javascript.
- ❑ Javascript là một ngôn ngữ kịch bản (script) để viết kịch bản cho phía client.



Giới thiệu

❑ Cấu trúc của đoạn Javascript:

```
<Script language="JavaScript"  
    type="text/javascript">
```

```
    Các lệnh Javascript;
```

```
    ...
```

```
</script>
```

Giới thiệu

□ Cách sử dụng và khai báo:

- Chèn trực tiếp đoạn code Javascript vào tập tin HTML
- Viết một tập tin Javascript riêng sau đó kết nối với trang web

Giới thiệu

❑ Chèn trực tiếp đoạn code Javascript vào tập tin HTML:

```
<HTML>
```

```
<HEAD>
```

```
<script language="javascript"  
  type="text/javascript">  
  document.write("What is your name? ");
```

```
...
```

```
</script>
```

```
</HEAD>
```

```
<BODY>
```

```
    Nội dung của trang
```

```
</BODY>
```

```
</HTML>
```

Giới thiệu

Chú ý:

- Có thể chèn đoạn code `<script> </script>` vào bất kỳ vị trí nào trong trang HTML.
- Có thể đặt trong phần `<head></head>` hoặc trong `<body> </body>` tuy nhiên ta nên đặt trong cặp tag `<head>` để dễ kiểm soát mã lệnh và cũng dễ sửa đổi chương trình.

Giới thiệu

□ Viết một tập tin Javascript riêng sau đó kết nối với trang web:

- Bước 1: Soạn thảo file Javascript có phần mở rộng .js
- Bước 2: Nhúng file .js và trang html:

```
<script SRC="fileJavascript.js"  
    language="javascript"  
    type="text/javascript">  
</script>
```


Giới thiệu

❑ Môi trường viết JAVASCRIPT

- Frontpage
- Notepad
- Visual InterDev
- Dreamweaver

BIẾN VÀ DỮ LIỆU

Biến:

- ❑ Khái niệm: lưu thông tin giá trị, không xác định kiểu dữ liệu.
- ❑ Khai báo: **var tên_biến;**

Ví dụ:

```
var x;
```

```
var a, b=100;
```

```
var name="KHOA CNTT";
```

```
b= "Đại Học Công Nghiệp";
```

BIẾN VÀ DỮ LIỆU

- ❑ Quy tắc đặt tên biến:
 - Tên biến gồm các chữ cái và số, không dùng các ký tự đặc biệt như: (, [, { , # , &)
 - Tên biến phải bắt đầu bằng ký tự hoặc ký tự
 - Không chứa khoảng trắng, **tên biến phải gọi nhớ**
 - Không trùng với từ khoá của JavaScript

BIẾN VÀ DỮ LIỆU

❏ Có 2 xuất biến:

```
document.write(tên_biến) ;
```

```
document.getElementById("ID").innerHTML=biến ;
```

Ví dụ:

```
var a="100";
```

```
var ten="Minh";
```

```
document.write(a) ;
```

```
document.getElementById("tt").innerHTML=ten ;
```

BIẾN VÀ DỮ LIỆU

Dữ liệu:

- **Kiểu số:** số thập phân, số nguyên
- **Kiểu chuỗi:** một biến kiểu chuỗi đặt trong cặp dấu nháy đôi (" ") hoặc đơn (' ')

□ Ví dụ:

```
var  s1, s2, s3 ;
```

```
s1=15;
```

```
s2='Hello World' ;
```

BIẾN VÀ DỮ LIỆU

➤ **Kiểu Boolean:** Là dữ liệu chỉ có 2 giá trị False hoặc True

Ví dụ:

```
var flag;  
  
flag=true ;
```

➤ **Kiểu Null:** biến không gán cho giá trị

Toán tử số học

Toán Tử	Chức Năng	Ví dụ	Kết quả
+	Cộng	x=2; x+2	4
-	Trừ	x=2; 5-x	3
*	Nhân	x=4; x*5	20
/	Chia	15/5; 5/2	3; 2.5
%	Lấy phần dư	5%2; 10%2	1; 0
++	Tăng giá trị lên 1	x=5; x++	x=6
--	Giảm giá trị xuống 1	x=5; x--	x=4

Toán tử gán

Tóan Tử	Ví dụ	Tương đương
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$

Toán tử so sánh

Tóan Tử	Chức Năng	Ví dụ
<code>==</code>	bằng	<code>5==8</code> returns false
<code>!=</code>	Không bằng	<code>5!=8</code> returns true
<code>></code>	lớn hơn	<code>5>8</code> returns false
<code><</code>	nhỏ hơn	<code>5<8</code> returns true
<code>>=</code>	lớn hơn hoặc bằng	<code>5>=8</code> returns false
<code><=</code>	nhỏ hơn hoặc bằng	<code>5<=8</code> returns true

Toán tử so sánh

Tóan Tử	Chức Năng	Ví dụ
&&	và	$x = 6; y = 3;$ $(x < 10 \ \&\& \ y > 1)$ returns true
	hoặc	$x = 6; y = 3$ $(x == 5 \ \ y == 5)$ returns false
!	not	$x = 6; y = 3;$ $!(x == y)$ returns true

Toán tử nối chuỗi

□ **+** : Là phép toán nối hai chuỗi với nhau

Ví dụ:

```
<html>
```

Chào bạn Mai

```
<script>
```

```
    var txt1="Chào bạn ";
```

```
    var ten="Mai";
```

```
    document.write ('<h2>'+txt1+ten+'</h2>') ;
```

```
</script>
```

```
</html>
```

Một số ký tự đặc biệt:

`\n` (new line)

`\t` (tab)

`\b` (BackSpace)

`\&` (dấu &)

`\”(“)`

Các lệnh trong Javascript

❑ *Statement (câu lệnh):*

Statement được kết thúc bằng dấu chấm phẩy(;)

ví dụ: **document.write("Hello") ;**

❑ *Block (Khối lệnh):*

Block là tập hợp nhiều statements được bao bọc bởi cặp dấu { ... }

Các lệnh trong Javascript

❏ *Comment (Lời chú thích):*

JavaScript hỗ trợ 2 loại comment:

- Chú thích trên một dòng: dùng cặp dấu //
- Chú thích trên nhiều dòng: dùng cặp dấu `/*...*/`

```
document.write("Welcome to  
JavaScript");
```

//câu lệnh xuất dữ liệu

Các câu lệnh cơ bản

Xuất dữ liệu ra trang Web:

Cách 1:

document.write(value) ;

document.writeln(value) ; //xuống dòng

❑ Trong đó **value**:

- Dữ liệu chuỗi: luôn đặt “ ”
- Biến: không có dấu “ ”
- Chuỗi và biến : dùng toán tử “+” nối biến và chuỗi

Cách 2: **Id.innerHTML="Giá trị";**

Các câu lệnh cơ bản

Ví dụ:

```
document.write("Chúc mừng năm mới" );  
document.write("String " + variable );  
noidung.innerHTML= "Chúc mừng năm mới"
```

Chú ý:

*Xuất tag HTML thì cặp tag đó cũng phải đặt trong cặp dấu
" "*

```
document.write("<table>");  
document.write("<tr>");
```


<BODY>

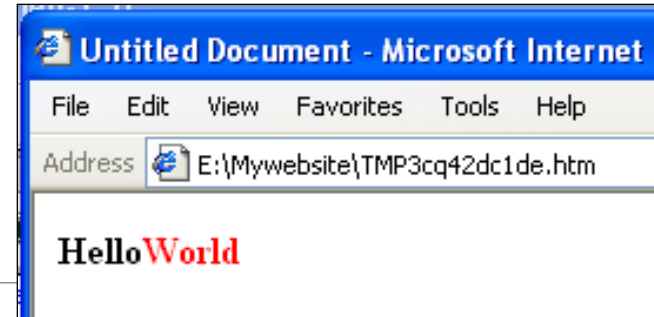
<Script Language="JavaScript">

document.write("<H1>Hello<H1>");

document.write("World");

</Script>

</BODY>



<BODY>

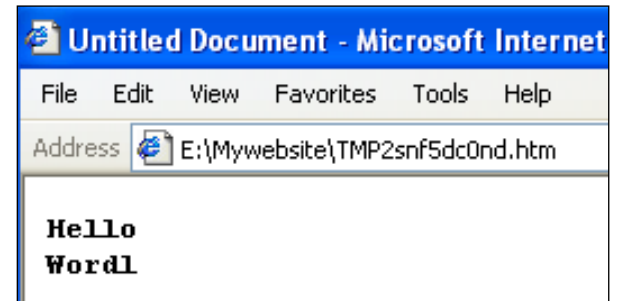
<Script Language="JavaScript">

document.writeln("Hello");

document.writeln("World");

</Script>

</BODY>



HÀM TRONG JAVASCRIPT

Hàm là một đoạn chương trình có thể được sử dụng nhiều lần trong một chương trình để thực hiện một tác vụ nào đó.

Xây dựng hàm:

```
function tên_hàm(các tham số)  
{  
    Khai báo các biến;  
    Các câu lệnh;  
    [return[giá trị /biểu thức]];  
}
```

HÀM TRONG JAVASCRIPT

Cách gọi hàm

Hàm sẽ không thực hiện cho đến khi nó được gọi.

- ❑ `Tên_hàm(đối số) ;`
- ❑ `Tên_hàm() ;`

HÀM TRONG JAVASCRIPT

- ❑ Đối với hàm không có giá trị trả về :

Tên_hàm(đối số) ;

Ví dụ: xuất(SV);

- ❑ Đối với hàm có giá trị trả về :

Tên_biến= Tên_hàm(đối số) ;

Ví dụ: kq= TínhDT(chieudai, chieurong) ;

HÀM TRONG JAVASCRIPT

```
function Display(user , pwd)
{
    document.write("UserName của bạn
    là:" + user) ;

    document.write("Password của bạn
    là:" + pwd) ;

    [return] ;
}
```

Lời gọi hàm:

Display(txt1,txt2) ;

HÀM TRONG JAVASCRIPT

```
function total(a,b)
{
    var c=0;
    c=a+b;
    return c;
}
```

Lời gọi hàm:
`kq=total(s1,s2);`

CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

alert(): dùng hiển thị một 1 message chứa 1 button OK

Cú pháp:

alert("nội dung Message") ;

ví dụ:

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      alert("Hello World")
    </script>
  </body>
</html>
```



CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

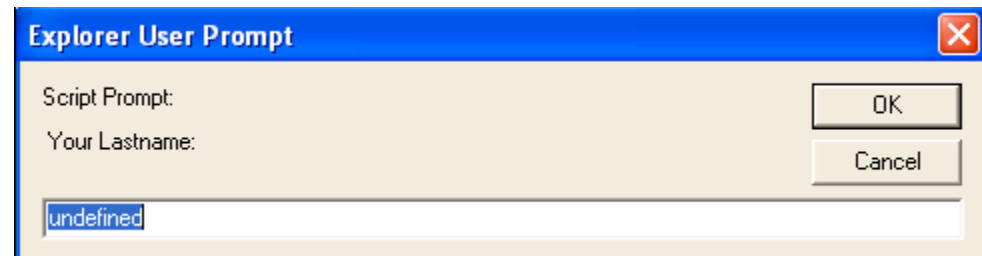
prompt(): Dùng để nhập dữ liệu từ trường text box của message. Hàm trả về giá trị *chuỗi* mà người sử dụng nhập vào

Cú pháp:

Tên_biến= prompt("message", giá trị) ;

CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      a=prompt("Your Lastname:");
      b=prompt("Your FirstName");
      document.write("Your FullName is :"+ a + ' ' + b)
    </script>
  </body>
</html>
```



CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

confirm(): Hiển thị một thông báo có hai chọn lựa OK và Cancel. Hàm trả về giá trị là true khi người nhấn OK và false khi nhấn Cancel

Cú pháp:

Tên_biến=confirm("Chuỗi Message") ;

CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

```
<html>
  <head><title>Function</title></head>
  <body>
    <script>
      a=prompt("nhap so a :");
      b=prompt("nhap so b");
      c=confirm( a +' lon hon '+ b+'?')
      if(c==true)
        document.write( a +" > "+b )
      else
        document.write( a +" < "+b )
    </script>
  </body>
</html>
```



CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

- ❑ **eval(chuỗi số)** : trả về giá trị số của một chuỗi số
- ❑ **isNaN(str)** :
 - True nếu str là chuỗi
 - False nếu str là chuỗi số.

CÁC HÀM THÔNG DỤNG TRONG JAVASCRIPT

Ví dụ:

```
var a=prompt("nhap so thu 1", 100); → "100";
```

```
    a=eval(a); → 100
```

```
var ten=prompt("Nhap ten ban", "Minh");
```

```
var flag=isNaN(ten); → true
```

```
var soluong=prompt("Nhap so luong");
```

```
var flag=isNaN(soluong); → false
```

CẤU TRÚC ĐIỀU KIỆN

Nội dung

❑ If ... Else

❑ Switch ... case

❑ For

❑ Do ...while

❑ While...

Ví dụ: Viết chương trình nhập 3 cạnh của tam giác sau đó xuất ra màn hình đó là tam giác gì?

<Body>

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

```
    a=eval(prompt("Nhap canh a")) ;
```

```
    b=eval(prompt("Nhap canh b")) ;
```

```
    c=eval(prompt("Nhap canh c")) ;
```

```
if(a==b && b==c && c==a)
```

```
    Tam giác đều ;
```

```
else
```

```
    if(a==b || b==c || c==a)
```

```
        Tam giác cân
```

```
else
```

```
    Tam giác thường
```

```
</script>
```

</Body>

Ví dụ:

<body>

<script>

```
t=prompt("nhap thang: ");
```

```
switch(eval(t))
```

```
{
```

```
    case 1: case 3: case 5: case 7: case 8 : case 10: case 12:
```

```
        alert("Thang "+ t+ " co 31 ngay");
```

```
        break;
```

```
    case 2:
```

```
        alert("Thang "+t + " co 28 ngay");
```

```
        break;
```

```
    case 4: case 6: case 9: case 11:
```

```
        alert("Thang "+t +" co 30 ngay");
```

```
        break;
```

```
    default:
```

```
        alert("Khong co thang nay");
```

```
}
```

</script>

</body>

❑ Ví dụ: Viết chương trình tạo một table m dòng n cột (m, n nhập tùy ý)

<body>

<Script language="javascript">

```
var n, m, i, j;
m=prompt("Nhap so dong");
n=prompt("Nhap so cot");
document.write("<table width=50% border=1>");
for(i=1;i<=m;i++)
{
    document.write("<tr>");
    for(j=1;j<=n;j++)
        document.write("<td>" + i + j + "</td>");
    document.write("</tr>");
}
document.write("</table>");
```

</Script>

</body>

```
<script language= "JavaScript">
```

```
for (x=1; x<=10 ; x++)
```

```
{          y=x*25;
```

```
document.write("x =" + x + " ; y= " + y + "<BR>") ;
```

```
}
```

```
</script>
```

