

# Javascript

CLIENT -SIDE PROGRAMMING

# Giới thiệu

2

- Javascript là ngôn ngữ lập trình, nhẹ và phổ biến cho web.
- Tương tác với người dùng và làm web động hơn.
- Ngôn ngữ thông dịch với khả năng hướng đối tượng.
- Tên ban đầu là Livescript, xuất hiện năm 1995 với trình duyệt Netscap 2.0

# Client - Side JavaScript

3

- ▶ Script bao gồm và điều khiển tài liệu HTML
- ▶ Javascript được gọi khi submit form, click vào button, chọn link, hoặc những hành động khác của người dùng.

# Ưu điểm và giới hạn của Javascript

4

Ưu điểm	Giới hạn
<ul style="list-style-type: none"><li>• Ít tương tác với server</li></ul>	Không cho phép làm việc với files
<ul style="list-style-type: none"><li>• Phản hồi nhanh với user (feedback)</li></ul>	Không phải là ứng dụng mạng
<ul style="list-style-type: none"><li>• Tăng tính tương tác (mouse, keyboard)</li></ul>	Không xử lý đa luồng, đa nhiệm
<ul style="list-style-type: none"><li>• Giao diện phong phú (drag drop, slider)</li></ul>	



# Cú pháp

- ▶ Sử dụng thẻ script
  - ▶ Language: javascript
  - ▶ Type: text/javascript
- ▶ Đặt bất cứ vị trí nào trong tài liệu:
  - ▶ Head
  - ▶ Body
  - ▶ External file và phải được khai báo trong phần head.

```
<body>  
<script type = "text/javascript">  
document.write("My first JavaScript");  
</script>  
</body>
```

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

## Lưu ý viết script

- ▶ Cuối mỗi lệnh có hoặc không có dấu ; (semicolon)
- ▶ Phân biệt chữ hoa, thường (case sensitivity)
- ▶ Ghi chú: // (dòng)  
/\* đoạn \*/  
<!-- đoạn //-->

# VD: Viết javascript

- ▶ Viết trong head
  - ▶ function sayHello()
  - ▶ Gọi function khi click vào button

```
<html>
<head>
<script type="text/javascript">
<!--
function sayHello() {
    alert("Hello World")
}
//-->
</script>
</head>
<body>
Click here for the result
<input type="button" onclick="sayHello()" value="Say Hello" />
</body>
</html>
```

## VD2: Viết javascript

- ▶ Viết trong body
  - ▶ Cần kết quả ở đâu thì viết ở đó.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
<!--
document.write("Hello World")
//-->
</script>
<p>This is web page body </p>
</body>
</html>
```



## VD3: Viết javascript

- ▶ Viết trong tệp script riêng:
  - Tạo trang js mới filename.js
  - Viết function sayHello.

- ▶ Khai báo script trong head

```
<script  
type="text/javascript  
"
```

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

9

```
function sayHello() {  
    alert("Hello World")  
}
```

```
<html>  
<head>  
<script type="text/javascript" src="filename.js" ></script>  
</head>  
<body>  
.....  
</body>  
</html>
```

# Biến và kiểu dữ liệu cơ sở

- ▶ Kiểu dữ liệu cơ sở: number (**int**, **float**), string, boolean, null, undefined, object, array.
- ▶ Khai báo biến: từ khóa **var**  
VD: **var** myname, id
- ▶ Javascript không bắt buộc xác định kiểu dữ liệu cho biến. Biến có thể lưu nhiều kiểu dữ liệu.
  - ▶ Biến toàn cục (phạm vi toàn tài liệu)
  - ▶ Biến cục bộ (phạm vi trong function)

# Biến và kiểu dữ liệu cơ sở

11

```
<script type = "text/javascript">
  //
  //from addNumbers.html

  var x = 5;
  var y = 3;
  var sum = x + y;

  alert(x + " plus " + y + " equals " + sum);

  //]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="532 310 901 506" data-label="Image"><img alt="Screenshot of a JavaScript alert dialog box titled '[JavaScript Application]'. The dialog displays a warning icon (a triangle with an exclamation mark) and the text '5 plus 3 equals 8'. An 'OK' button is located at the bottom right of the dialog."/>A screenshot of a JavaScript alert dialog box. The window title is "[JavaScript Application]". Inside the dialog, there is a warning icon (a triangle with an exclamation mark) on the left and the text "5 plus 3 equals 8" on the right. At the bottom center, there is an "OK" button.</div>
```

# Chuyển biến về đúng kiểu dữ liệu

12

<b>Table 1-1                      Variable Conversion Functions</b>				
<i>Function</i>	<i>From</i>	<i>To</i>	<i>Example</i>	<i>Result</i>
<code>parseInt()</code>	String	Integer	<code>parseInt("23")</code>	23
<code>parseFloat()</code>	String	Floating point	<code>parseFloat("21.5")</code>	21.5
<code>toString()</code>	Any variable	String	<code>myVar.toString()</code>	<i>varies</i>
<code>eval()</code>	Expression	Result	<code>eval("5 + 3")</code>	8
<code>Math.ceil()</code>	Floating point	Integer	<code>Math.ceil(5.2)</code>	6
<code>Math.floor()</code>			<code>Math.floor(5.2)</code>	5
<code>Math.round()</code>			<code>Math.round(5.2)</code>	5

# Arithmetic Operators

13

Symbol	Description	Example (x = 10)	Result
+	Addition	x+5	15
-	Subtraction	x-2	8
*	Multiplication	x*3	30
/	Division	x/2	15
%	Modulus (division remainder)	x%3	1
++	Increment (increments the variable by 1 — this technique is often used in counters)	x++	11
--	Decrement (decreases the variable by 1)	x--	9



# Assignment Operators

14

total = a + b;

Các toán tử viết tắt

Symbol	Example Using Shorthand	Equivalent Without Shorthand
+=	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

# Comparison Operators

15

Operator	Description	Example
==	Equal to	1==2 returns false 3==3 returns true
!=	Not equal to	1!=2 returns true 3!=3 returns false
>	Greater than	1>2 returns false 3>3 returns false 3>2 returns true
<	Less than	1<2 returns true 3<3 returns false 3<1 returns false
>=	Greater than or equal to	1>=2 returns false 3>=2 returns true 3>=3 returns true
<=	Less than or equal to	1<=2 returns true 3<=3 returns true 3<=4 returns false

# Logical or Boolean Operators

16

Operator	Name	Description	Example (where x=1 and y=2)
&&	And	Allows you to check if both of two conditions are met	(x < 2 && y > 1) Returns <code>true</code> (because both conditions are met)
??	Or	Allows you to check if one of two conditions are met	(x < 2 ?? y < 2) Returns <code>true</code> (because the first condition is met)
!	Not	Allows you to check if something is not the case	! (x > y) Returns <code>true</code> (because x is not more than y)

## Fixing the addInput code

- Convert dữ liệu nhập về đúng kiểu trước khi tính toán

```
var x = prompt("first number:");  
var y = prompt("second number:");  
var sum = parseFloat(x) + parseFloat(y);  
  
alert(x + " plus " + y + " equals " + sum);
```

# String Operator

- Ghép 2 string: dùng kí hiệu dấu +

```
firstName = "Bob"  
lastName = "Stewart"  
name = firstName + lastName
```



# Nhập, xuất cơ bản

- ▶ Nhập dữ liệu từ hộp thoại:

```
a = prompt ("Nhập số a: ");
```

- ▶ Xuất kết quả ra hộp thoại thông báo:

```
alert ("Kết quả là " + a );
```

- ▶ In ra HTML

```
document.write ("Kết quả là " + a );
```

```
document.writeln ("<p>Kết quả là " + a + "</p>");
```

# Bài tập 1

20

1. Viết script nhập vào số nguyên  $a$ ,  $b$ . Tính và in ra tổng, hiệu, tích, thương.
2. Sửa lại bài 1 nhập 2 số  $a$ ,  $b$  từ hộp thoại.

# Conditional Statements

21

- ▶ Câu lệnh IF, IF ... ELSE
- ▶ Điều kiện là biểu thức logic

```
if (condition)
{
    code to be executed if condition is true
}
```

```
if (condition)
{
    code to be executed if condition is true
}
else
{
    code to be executed if condition is false
}
```

# IF, IF ELSE

22

## ► Ví dụ

```
<script type="text/JavaScript">
  date = new Date();
  time = date.getHours();
  if (time < 12)
    document.write('Good Morning');
</script>
```

```
<script type="text/JavaScript">
  date = new Date();
  time = date.getHours();
  if (time < 12) {
    document.write('Good Morning');
  }
  else {
    document.write('Good Afternoon');
  }
</script>
```



# Switch

23

## ► Rẽ nhánh với nhiều giá trị chọn

```
switch (expression)
{
case option1:
    code to be executed if expression is what is written in option1
    break;
case option2:
    code to be executed if expression is what is written in option2
    break;
case option3:
    code to be executed if expression is what is written in option3
    break;
default:
    code to be executed if expression is different from option1, option2,
    and option3
}
```



# Câu lệnh rẽ nhánh ngắn

24

```
variablename= (condition)?value1:value2
```

```
instruction= (color=="red")?"STOP": "CONTINUE"
```

# Loop

- ▶ While: Kiểm tra điều kiện đúng thì thực hiện câu lệnh
- ▶ Do.. While: Kiểm tra điều kiện sau (lặp ít nhất 01 lần)
- ▶ Số lần lặp không biết trước.

```
while (condition)
{
    code to be executed
}
```

```
do
{
    code to be executed
}
while (condition)
```

# Loop

26

## ► Ví dụ: While

```
<script type="text/JavaScript">
i = 1
while (i < 11) {
  document.write(i + " x 3 = " + (i * 3) + "<br />" );
  i ++
}
</script>
```

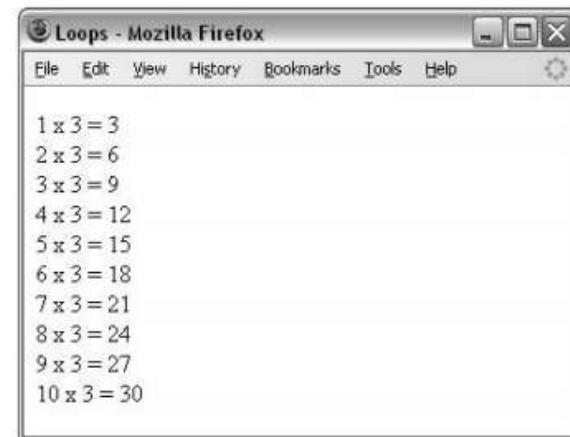


Figure 11-7

# For

- ▶ Lặp với số lần biết trước.
- a: giá trị khởi tạo
- b: điều kiện
- c: bước nhảy

```
for (a; b; c)
{
    code to be executed
}
```

```
for (i=0; i<11; i++) {
    document.write(i + " x 3 = " + (i * 3) + "<br />" );
}
```

# Break

- ▶ Hủy câu lệnh và thoát khỏi lặp.

```
for (i=0; /* no condition here */ ; i++) {  
    document.write(i + " x 3 = " + (i * 3) + "<br />" );  
    if (i == 100) {  
        break;  
    }  
}
```



# Function

## ► Khai báo từ khóa function

```
function calculateArea(width, height) {  
    area = width * height  
    return area  
}
```

## ► Gọi hàm bằng tên hàm và truyền tham số thật (nếu có)

- Gọi hàm khi có sự kiện
- Gọi hàm trong script

```
function cong(a,b)  
{  
    return a+b;  
}  
document.write(cong (5,10));
```

```
<input type="submit" onclick="alert(cong(5,10))" />
```