
Chương 3

JS - JavaScript

NỘI DUNG

- Giới thiệu về Javascript
- Nhúng Javascript vào trang web
- Kiểu dữ liệu & Các cú pháp Javascript
- Xử lý sự kiện
- DOM HTML với Javascript
- Ví dụ

Giới thiệu về JavaScript

- Script?
 - Ngôn ngữ kịch bản
 - Là một ngôn ngữ lập trình cho phép tạo ra sự điều khiển trong một ứng dụng khác.
 - Các ngôn ngữ kịch bản thường được thông dịch
- JS là ngôn ngữ Client-side script hoạt động trên trình duyệt của người dùng (client)
- Chia sẻ xử lý trong ứng dụng web. Giảm các xử lý không cần thiết trên server.
- Giúp tạo các hiệu ứng, tương tác cho trang web.

Giới thiệu về JavaScript

- Client-Side Script:

- Script được thực thi tại Client-Side (trình duyệt): Thực hiện các tương tác với người dùng (tạo menu chuyển động, ...) , kiểm tra dữ liệu nhập, ...

- Server-Side Script:

- Script được xử lý tại Server-Side, nhằm tạo các trang web có khả năng phát sinh nội dung động. Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về người dùng...

Giới thiệu về JavaScript

- Cung cấp sự tương tác với người dùng
 - Sự kiện do người dùng tạo ra: di chuột, click chuột...
 - Sự kiện do hệ thống tạo ra: thay đổi kích thước trang, tải trang...
- Thay đổi động nội dung
 - Thay đổi nội dung và vị trí các thành phần trên trang Web theo sự tương tác của người dùng
- Hợp lệ hóa dữ liệu
 - Kiểm tra xem dữ liệu có hợp lệ không trước khi nó được gửi (submit) đến Web Server để xử lý

Nhúng JavaScript vào trang web

Nhúng Javascript vào trang web

Định nghĩa script trực tiếp trong trang html:

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

Nhúng sử dụng script cài đặt từ 1 file .js khác:

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

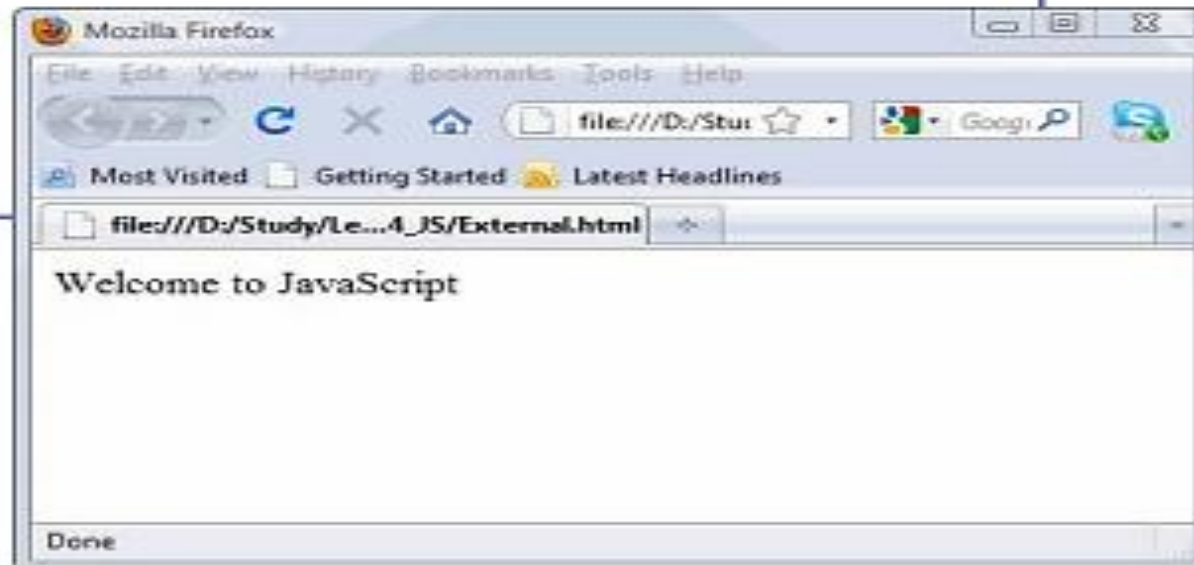
Nhúng JavaScript vào trang web

- Đặt giữa tag `<head>` và `</head>`: script sẽ thực thi ngay khi trang web được mở.
- Đặt giữa tag `<body>` và `</body>`: 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 client-script chèn vào trang không hạn chế.
- File JavaScript bên ngoài cho phép sử dụng lại nhiều lần và chỉnh sửa dễ dàng

Nhúng JavaScript vào trang web

Nhúng Javascript vào trang web

```
<html>
  <head>
    <script language="javascript">
      document.write("Welcome to JavaScript");
    </script>
  </head>
  <body>
  </body>
</html>
```



Kiểu dữ liệu và các cú pháp JavaScript

Các qui tắc chung

- Khối lệnh được bao trong dấu {}
- Mỗi lệnh nên kết thúc bằng dấu ;
- Cách ghi chú thích:
 - // Chú thích 1 dòng
 - /* Chú thích
nhiều dòng */

Biến số trong javascript

- Cách đặt tên biến

- Bắt đầu bằng một chữ cái hoặc dấu _
- A..Z, a..z, 0..9, _ : phân biệt HOA, Thường

- Khai báo biến

- Sử dụng từ khóa **var**

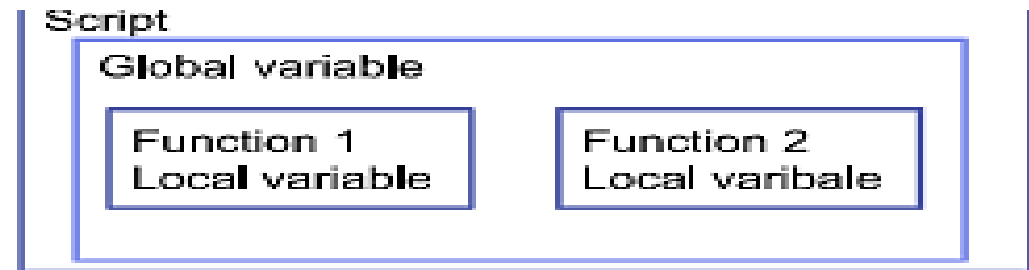
Ví dụ: **var count=10,amount;**

- Không cần khai báo biến trước khi sử dụng, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu tiên

Biến số trong javascript

Biến số trong Javascript

- Phạm vi của biến
 - Được xác định bởi nơi biến được khai báo
- Biến toàn cục
 - Khai báo bên ngoài hàm
 - Được truy cập từ mọi nơi trong kịch bản
 - Khai báo không cần từ khóa var
- Biến cục bộ
 - Khai báo bên trong hàm
 - Phạm vi trong hàm khai báo
 - Bắt buộc phải có từ khóa var



Ví dụ phạm vi biến

```
<html>
  <head>
    <title>Trang web</title>
    <meta charset="UTF-8">
    <script>
      var a=10,b=20; //toàn cục
      function tong() {
        return (a+b);
      }
      function tru() {
        a=5; //cục bộ trong hàm
        return (a-b);
      }
    </script>
  </head>
  <body>
    <p id="tong"></p>
    <p id="tru"></p>
    <script>
      document.getElementById("tong").innerHTML = tong();
      document.getElementById("tru").innerHTML = tru();
    </script>
  </body>
</html>
```

30

-15

Kiểu dữ liệu trong javascript

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 <code>new</code>
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>

Đổi kiểu dữ liệu

- Biến tự đổi kiểu dữ liệu khi giá trị mà nó lưu trữ thay đổi

Ví dụ:

```
var x = 10;           // x kiểu Number  
x = "hello world !"; // x kiểu String
```

- Có thể cộng 2 biến khác kiểu dữ liệu

Ví dụ:

```
var x;  
x = "12" + 34.5; // KQ: x = "1234.5"
```

- Hàm parseInt(...), parseFloat(...) : Đổi KDL từ chuỗi sang số.
 - Chuyển không thành công: "NaN"
-

Đổi kiểu dữ liệu

```
<script language="javascript">
    var x = 10;
    document.write(x);
    x=true;
    x = "Hello World";
    document.write(x);
    x = "12"+4.3;
    document.write(x);
    x = parseInt("-32");
    document.write(x);
</script>
```

10 Hello world 124.3 -32

Phép toán số học

Phép toán	Mô tả	Ví dụ
+	Cộng	$A = 5 + 8$
-	Trừ	$A = 8 - 5$
/	Chia	$A = 20 / 5$
%	Số dư	$10 \% 3 = 1$
++	Tăng 1	<code>++ x</code> và <code>x ++</code>
--	Giảm 1	<code>-- x</code> và <code>x --</code>
-	Đảo dấu	nếu $A = 5$ thì $-A = -5$

Phép toán so sánh

Phép toán	Mô tả so sánh	Ví dụ (x=5)	Kết quả
<code>==</code>	Bằng giá trị	<code>x == 5</code>	<code>true</code>
<code>===</code>	Bằng giá trị và kiểu	<code>x === "5"</code>	<code>false</code>
<code>!=</code>	Không bằng	<code>x != 4</code>	<code>true</code>
<code>></code>	Lớn hơn	<code>x > 5</code>	<code>false</code>
<code>>=</code>	Lớn hơn hoặc bằng	<code>x >= 5</code>	<code>true</code>
<code><</code>	Nhỏ hơn	<code>x < 4</code>	<code>false</code>
<code><=</code>	Nhỏ hơn hoặc bằng	<code>x <= 5</code>	<code>true</code>

Phép toán logic

Phép toán	Mô tả	Ví dụ (x=6, y=3)	Kết quả
&&	And	(x<10) && (y>1)	true
	Or	(x<1) (y>5)	false
!	Not	! (x==y)	true

Phép toán định lượng

- Phép toán điều kiện

Cú pháp

```
tênbiến = điềukiện ? giátrị1:giátrị2;
```

Ví dụ:

```
status = (diem >= 5) ? “đạt”: “thi lại”;
```

Phép toán định lượng

- `typeof`

- Trả về xâu xác định kiểu dữ liệu của biến

Ví dụ:

```
var x = 5;  
document.write(typeof(x));  
  
=> number
```

Hàm trong Javascript

- Dạng thức khai báo chung:

```
function Tên_hàm(thamsol, thamso2,...)  
{  
    .....  
}
```

- Hàm có giá trị trả về:

```
function Tên_hàm(thamsol, thamso2,...)  
{  
    .....  
    return (value);  
}
```

Hàm trong Javascript

- Ví dụ:

```
function Sum(x, y)
{
    tong = x + y;
    return tong;
}
```

- Gọi hàm:

```
var x = Sum(10, 20);
```

Câu lệnh 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;
```


Câu lệnh Switch

Câu lệnh switch

switch (***expression***)

{

case label :

statementlist

case label :

statementlist

 ...

default :

statement list

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")

}

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;  
}
```

Vòng lặp for sử dụng break

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var text = "";
for (var i = 0; i < 10; i++) {
  if (i === 3) { break; }
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

The number is 0
The number is 1
The number is 2

Vòng lặp for sử dụng continue

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var text = "";
for (var i = 0; i < 10; i++) {
  if (i === 3) { continue; }
  text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

The number is 0
The number is 1
The number is 2
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9

Vòng lặp for in object

```
for (key in object) {  
    // code block to be executed  
}
```

```
<!DOCTYPE html>  
<html>  
<body>  
<p id="demo"></p>  
<script>  
const person = {fname:"John", lname:"Doe", age:25};  
var txt = "";  
for (var x in person) {  
    txt += person[x] + " ";  
}  
document.getElementById("demo").innerHTML = txt;  
</script>  
</body>  
</html>
```

John Doe 25

Vòng lặp for in Arrays

```
for (variable in array) {  
    code  
}
```

```
<!DOCTYPE html>  
<html>  
<body>  
<p id="demo"></p>  
<script>  
const numbers = [45, 4, 9, 16, 25];  
var txt = "";  
for (var x in numbers) {  
    txt += numbers[x] + "<br>";  
}  
document.getElementById("demo").innerHTML = txt;  
</script>  
  
</body>  
</html>
```

45
4
9
16
25

Array.forEach()

Phương thức `forEach()` gọi một hàm (một hàm gọi lại) một lần cho mỗi phần tử mảng.

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
const numbers = [45, 4, 9, 16, 25];
let txt = "";
numbers.forEach(myFunction);
document.getElementById("demo").innerHTML = txt;
function myFunction(value, index, array) {
  txt += value + " - " + index + " - " + array + "<br>";
}
</script>
</body>
</html>
```

```
45 - 0 - 45,4,9,16,25
4 - 1 - 45,4,9,16,25
9 - 2 - 45,4,9,16,25
16 - 3 - 45,4,9,16,25
25 - 4 - 45,4,9,16,25
```

Vòng lặp For Of

```
for (variable of iterable) {  
    // code block to be executed  
}
```

Câu lệnh JavaScript for lặp qua các giá trị của một đối tượng có thể lặp lại.

Vòng lặp For Of array

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
const cars = ["BMW", "Volvo", "Mini"];
var text = "";
for (var x of cars) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

BMW
Volvo
Mini

Vòng lặp For Of String

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var language = "JavaScript";

var text = "";
for (var x of language) {
    text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

J
a
v
a
S
c
r
i
p
t

Vòng lặp For Of Set (tập hợp)

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
// Create a Set
const letters = new Set(["a","b","c"]);
// List all Elements
var text = "";
for (var x of letters) {
    text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

a
b
c

Vòng lặp For Of map

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
// Create a Map
const fruits = new Map([
  ["apples", 500],
  ["bananas", 300],
  ["oranges", 200]
]);
// List all entries
var text = "";
for (var x of fruits) {
  text += x + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

apples,500
bananas,300
oranges,200

Vòng lặp while, do...while

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

Ví dụ:

```
var i = 9, total = 0;
while (i < 10)
{
    total += i * 3 + 5;
    i = i + 5;
}
```

```
do
{
    statement
}while (expression);
```

Ví dụ:

```
var i = 9, total = 0;
do
{
    total += i * 3 + 5;
    i = i + 5;
} while (i > 10);
```

While sử dụng break

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var text = "";
var i=0;
while(i<10)
{
    if (i === 3)
    { break;
    }
    text += "The number is " + i + "<br>";
    i++;
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
</html>
```

The number is 0
The number is 1
The number is 2

Hộp thông báo

- Sử dụng các phương thức: confirm, alert, prompt
- Phương thức alert()

Cú pháp

```
alert ("thông_báo");
```

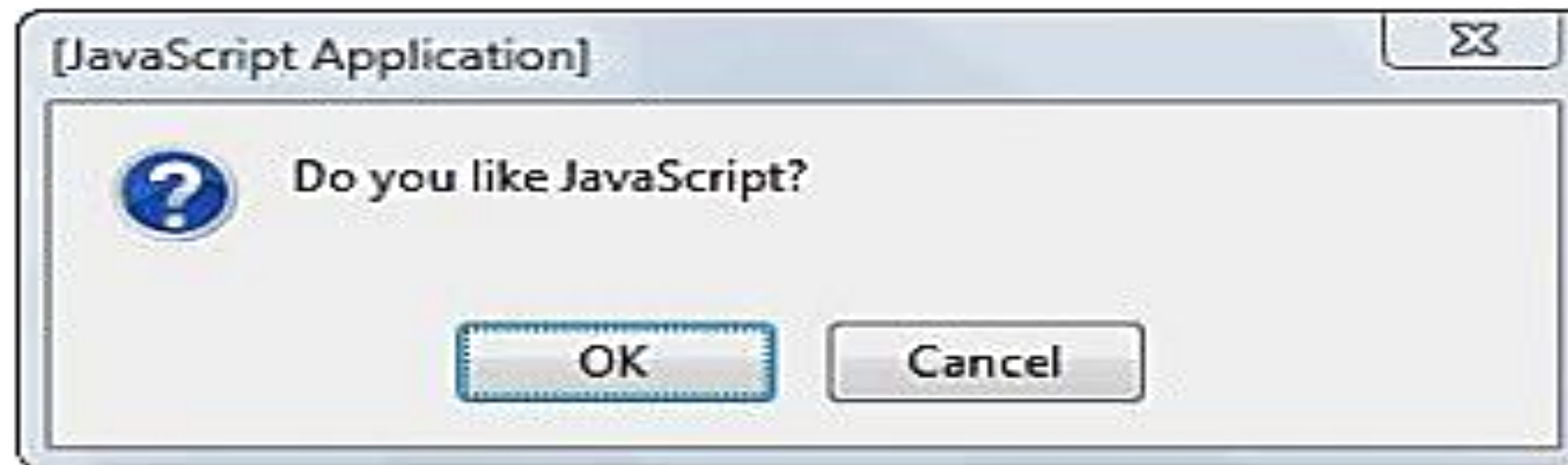


Hộp thông báo

- Phương thức confirm()

Cú pháp

```
ret = confirm ("thông_báo");
```



Hộp thông báo

- Phương thức prompt()

Cú pháp

```
value = prompt("thông_báo",  
giá_trị_mặc_định);
```



Hộp thông báo

```
<script language="javascript">
    alert("Welcome to JavaScript");
    var value = prompt("What's your
                        name?", "JS");
    alert("Hello "+value);
    var ret = confirm("Do you like
                      JavaScript?");
    alert("You pressed "+ret);
</script>
```

Hộp thông báo



Viết chú thích

- Viết chú thích trên một dòng: //
- Viết chú thích trên nhiều dòng: /* và */

```
/*Todo: Example in MessageBox  
include: alert(), confirm(),  
prompt()  
*/  
var value = prompt("What's your  
name?", "JS"); //get your name
```

Các kí tự đặc biệt

- Vấn đề:

```
var txt="He is very "intelligence";
```

- Các kí tự đặc biệt trong JavaScript:

“ ’ & \ n r t b f

- Để sử dụng: thêm dấu \ phía trước

```
var txt=
```

```
"He is very \"intelligence\"";
```

Xử lý sự kiện

Giới thiệu về sự kiện

- Sự kiện (event)
 - Là các hành động có thể được thể hiện bằng JavaScript
 - Mỗi thành phần trên trang Web đều có những sự kiện xác định, có thể kích hoạt một script
- Ví dụ
 - Sự kiện click chuột
 - Trang web hoặc ảnh được tải
 - Nhập dữ liệu vào một trường trên form

Các sự kiện thông dụng

- Chú ý:

- Sự kiện thường được kết hợp với hàm. Hàm sẽ không thực thi chừng nào sự kiện chưa xuất hiện
- Cú pháp

```
sự_kiện = "tên_hàm () ;"
```

hoặc

```
sự_kiện = "tên_hàm (ds_tham_số) ;"
```


Các sự kiện thông dụng

- Các sự kiện được hỗ trợ bởi hầu hết các đối tượng
 - `onClick`
 - `onFocus`
 - `onChange`
 - `onBlur`
 - `onMouseOver`
 - `onMouseOut`
 - `onMouseDown`
 - `onMouseUp`
 - `onLoad`
 - `onSubmit`
 - `onResize`
 -

Xử lý các sự kiện cho các thẻ HTML

- Cú pháp 1:

`<TAG eventName = "JavaScript Code">`

```
<!DOCTYPE html>
<html>
<body>
  <input type="button" name="" value="Show" onclick="window.open('https://www.google.com');"/>
</body>
</html>
```

Xử lý sự kiện bằng function

```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function cong(a,b) {
        document.write(a+b);
      }
    </script>
  </head>
  <body onload="cong(20,30);">
</body>
</html>
```

Một số sự kiện thông dụng

Sự kiện	Ý nghĩa
onMouseUp	nút chuột được nhả
onMouseDown	ấn chuột
onMouseOver	chuột di chuyển qua vùng
onMouseOut	chuột di chuyển khỏi vùng
onMouseMove	chuột được di chuyển
onclick	click chuột
ondblclick	click đúp chuột
onFocus	Đối tượng được sử dụng (đặt con trỏ)
onBlur	Đối tượng không còn được sử dụng
onChange	Đối tượng được thay đổi (nhập dữ liệu)
onSubmit	Xác nhận tất cả dữ liệu trên form

Ví dụ: Onclick Event

```
<script>
  function compute(frm) {
    var x=frm.expr.value;
    result.innerHTML=x*x;
  }
</script>
```

```
<body>
  <form name="frm" >
    x= <input type="text" name="expr" value="" />
    <br>
    <br>
    <input type="button" value="calculate" onclick="compute(this.form);" />
    <br>
    <br>
    x*x= <b id="result"></b>
  </form>
</body>
```

x=

calculate

x*x=

x=

calculate

x*x= **16**

Ví dụ: Onclick Event

```
<!DOCTYPE html>
<html>

<body>
  <input type="text" value="" onblur="(document.bgColor='red')" onfocus="(document.bgColor='brown')" />
</body>
</html>
```

