



NỀN TẢNG CỦA CÚ PHÁP NGÔN NGỮ JavaScript

Session 6



MỤC TIÊU

- Mô tả về JavaScript
- Nắm bắt nền tảng cú pháp của JavaScript



JavaScript là gì?

- **JavaScript là ngôn ngữ kịch bản dùng để tạo các client-side scripts và server-side scripts.**
- **JavaScript làm cho việc tạo các trang Web động và tương tác dễ dàng hơn**
- **JavaScript làm một ngôn ngữ kịch bản được hãng Sun Microsystems và Netscape phát triển.**
- **JavaScript được phát triển từ **Livescript**. Của Netscape**
- **Các ứng dụng client chạy trên một trình duyệt như Netscape Navigator hoặc Internet Explorer.**



Các hiệu ứng và các quy tắc của JavaScript

- **JavaScript có thể tăng cường tính động và tính tương tác của các website bằng cách sử dụng các hiệu ứng của nó.**
 - **Cung cấp sự tương tác người dùng**
 - **Thay đổi nội dung động**
 - **Xác nhận tính hợp lệ của dữ liệu**
- **Tương tự các ngôn ngữ khác, JavaScript cũng tuân theo các quy tắc ngữ pháp như:**
 - **Sử dụng quy tắc chữ hoa**
 - **Using Pairs**
 - **Using Spaces**
 - **Using Comments**



Các công cụ của JavaScript và môi trường thực thi

- Các công cụ sinh mã JavaScript và giao phần mềm có giao diện phát triển ứng dụng IDE được sử dụng giúp tự động tạo ra các đoạn mã JavaScript code. Một vài chức năng được sinh mã:
 - Dialog Box
 - Pop – up Menu Builder
 - Remotes
- Môi trường thực thi
 - Các Scripting ở Client
 - Java Script trên Web Server



Nhúng JavaScript vào trang Page

- JavaScript có thể chèn vào một tài liệu HTML theo những cách sau :

- sử dụng thẻ SCRIPT:

```
<script language="JavaScript">  
  <!--  
    JavaScript statements;  
  //-->  
</script>
```

- sử dụng một file JavaScript ở ngoài

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

- Sử dụng các biểu thức JavaScript trong các giá trị thuộc tính của thẻ
 - Sử dụng JavaScript trong các trình điều khiển sự kiện

Chương trình dùng Msg box và phương thức write

Ví dụ:

```
<HTML>
```

```
<HEAD>
```

```
<SCRIPT LANGUAGE = "Javascript">
```

```
confirm ("Are you Sure?");
```

```
alert ("OK");
```

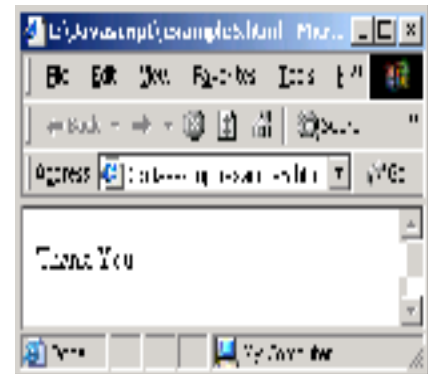
```
document.write(" Thank You !");
```

```
</SCRIPT>
```

```
</HEAD>
```

```
</HTML>
```

Kết quả:





Biến

- Biến là một vật chứa tham chiếu đến một vị trí ở bộ nhớ máy tính
- Nó được sử dụng để giữ giá trị và có thể thay đổi trong khi kịch bản thực thi
- Các biến tuân theo quy tắc đặt tên.
- Một biến được khai báo sử dụng từ khoá '**var**'.

ví dụ: `var A = 10;`

- Các biến có một phạm vi được xác định trong khi chúng khai báo trong script.
 - Biến toàn cục
 - Biến cục bộ
- Nguyên dạng là các giá trị không đổi được dùng trong script.



Các kiểu dữ liệu

- JavaScript có một tập các kiểu dữ liệu.
 - Số (number)
 - Giá trị logic (boolean)
 - Chuỗi (String)
 - Giá trị rỗng Null
- JavaScript phân biệt chữ hoa, chữ thường.
- Trong JavaScript, hai biến khác kiểu có thể kết hợp với nhau.

ví dụ: `A = " This apple costs Rs." + 5`

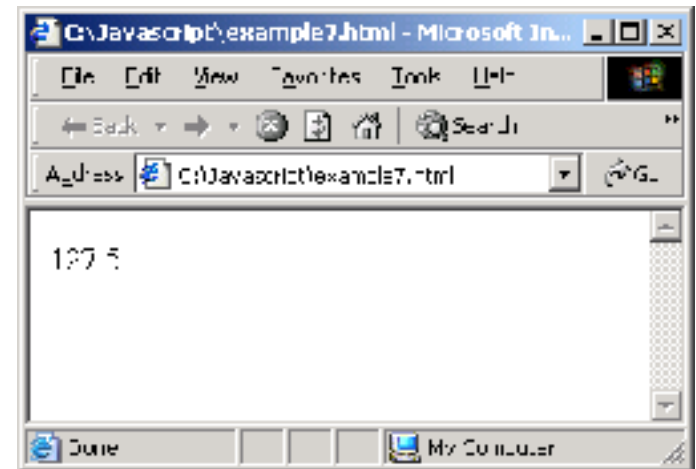
sẽ có kết quả là một chuỗi với giá trị là "This apple costs Rs. 5".

Kiểu dữ liệu – Ví dụ

Ví dụ:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Javascript">
    var A = "12" + 7.5;
    document.write(A);
</SCRIPT>
</HEAD>
</HTML>
```

Kết quả:





Các kiểu nguyên dạng (literal)

- Integer – là các hệ thống số thập phân, thập lục phân và nhị phân.
- Floating- point(số thực) – **Các số thập phân** có phần thập phân sử dụng "e" or ""E" và theo sau là các số nguyên.
- String – là một chuỗi rỗng hay chuỗi ký tự được đặt trong cặp ngoặc đơn hoặc ngoặc kép
- Boolean–Kiểu này có hai giá trị: True or False.
- null - Kiểu null chỉ có một giá trị: null. Null hàm ý không có dữ liệu.



Các toán tử

- Các toán tử xử lý một hoặc nhiều biến hoặc các giá trị (các toán hạng) và trả lại giá trị kết quả.
- JavaScript sử dụng cả hai toán tử một ngôi và hai ngôi.
- Các toán tử được phân loại phụ thuộc quan hệ chúng thực hiện như:
 - Toán tử số học
 - Toán tử so sánh
 - Toán tử logic
 - Toán tử chuỗi
 - Toán tử lượng giá
 - Mức ưu tiên của toán tử



Toán tử số học

- Các toán tử số học lấy hai toán tử (hoặc nguyên dạng hoặc các biến) và trả về một giá trị số.
- Toán tử số học gồm:
 - Cộng (+)
 - Trừ (-)
 - Nhân (/)
 - Chia (%)
 - Toán tử tăng (++)
 - Toán tử giảm (--)
 - Toán tử phủ định (-)



Toán tử so sánh

- Toán tử so sánh so sánh các toán hạng và trả về giá trị logic dựa trên sự so sánh có đúng hay không.
- Toán tử so sánh bao gồm:
 - So sánh bằng ($==$)
 - So sánh khác ($!=$)
 - So sánh lớn hơn ($>$)
 - So sánh lớn hơn hoặc bằng ($>=$)
 - So sánh nhỏ hơn ($<$)
 - So sánh nhỏ hơn hoặc bằng ($<=$)



Các toán tử logic

- Các toán tử logic dùng để kết hợp các so sánh trong một biểu thức điều kiện.
- Bao gồm:

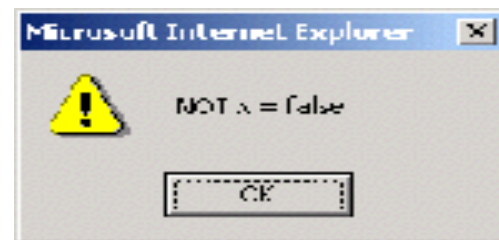
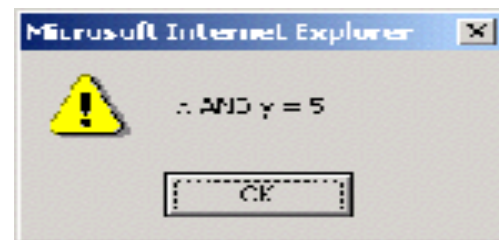
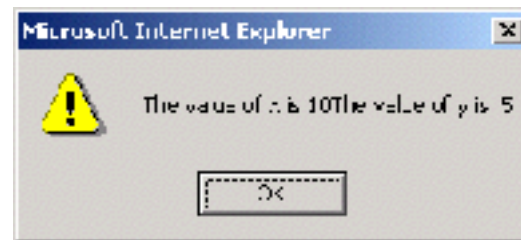
Operator	Value	Description
And (&&)	expr1 && expr2	Returns expr1 if it converts to false. Otherwise returns expr2.
Or ()	expr1 expr2	Returns expr1 if it converts to true. Otherwise returns expr2.
Not (!)	<u>!expr</u>	Returns false if the expression is true, and true if it is false.

Ví dụ về toán tử logic

Ví dụ:

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      var x = 10;
      var y = 5;
      alert ("The value of x is "
            + x + "The value of y is "
            + y);
      alert("x AND y = " + (x && y));
      alert("x OR y = " + (x || y));
      alert("NOT x = " + (!x));
    </SCRIPT>
  </HEAD>
</HTML>
```

Kết quả:





Toán tử chuỗi

- Toán tử chuỗi lấy các toán tử chuỗi như các toán hạng và tạo một chuỗi mới, kết quả là một chuỗi kết hợp các chuỗi con.

Ví dụ:

```
x = 'yellow';  
y = 'green';  
z = x + y + 'white'; which means z is  
"yellowgreenwhite"  
w = y + 9, which means w is "green9"
```



Toán tử lượng giá

- Các toán tử chỉnh bao gồm:

- **Toán tử điều kiện**

`(condition) ? trueVal : falseVal`

Gán một giá trị xác định vào một biến nếu điều kiện đúng, trường hợp còn lại thì gán vào biến còn lại.

ví dụ.

```
status = (age >= 18) ? "adult" : "minor"
```

- **Toán tử Typeof**

Toán tử typeof trả về một chuỗi chỉ ra kiểu của toán hạng.

ví dụ.

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

Mức ưu tiên của toán tử

- Khi có nhiều toán tử được tính toán trong một biểu thức, mức ưu tiên của toán tử xác định thứ tự thực hiện của toán tử trong biểu thức đó.
- Bảng sau liệt kê mức ưu tiên của các toán tử từ thấp đến cao:

Operator type	Individual operators
assignment	= += -= *= /= %/= <=> >>= >>>= &&= '== =
conditional	?:
logical-or	
logical-and	&&
bitwise-or	
bitwise-xor	^
bitwise-and	&
equality	== !=
relational	< <= > >=
bitwise shift	<< >> >>>
addition/subtraction	+ -
multiply/divide	* / %
negation/increment	~ - ++ -- <u>typedef void</u>



Biểu thức

- Biểu thức được dùng để xử lý và tính toán các biến trong các ngữ cảnh khác nhau.
- Một biểu thức là một tập giá trị hợp lý các nguyên dạng, các biến, và các toán tử và trả về một giá trị đơn.
- Các biểu thức của JavaScript bao gồm:
 - **Arithmetic:** tính toán một số
 - **Logical:** tính toán một giá trị logic
 - **String:** tính toán một chuỗi
- Các biểu thức kết hợp các biến và các nguyên dạng thông qua các toán tử.



Biểu thức quy tắc

- Một biểu thức quy tắc là một kiểu được xác định trong việc tìm kiếm tương ứng các ký tự kết hợp của một chuỗi.
- Các biểu thức quy tắc có thể dùng để tìm kiếm các mẫu ký tự trong một chuỗi nhập vào từ người sử dụng.
- Biểu thức quy tắc bao gồm:
 - Các mẫu đơn giản
 - Các ký tự đơn giản và ký tự đặc biệt
- Các biểu thức quy tắc có thể được tạo bằng một trong hai cách:
 - Sử dụng khởi tạo đối tượng
 - Gọi hàm khởi tạo của đối tượng RegExpt



Sử dụng biểu thức quy tắc

- Các phương thức sử dụng trong biểu thức quy tắc bao gồm:
 - Exec, Test, Match, Search, Replace, Split
- Cú pháp dùng phương thức:
`objectname.method = function_name`
- Cú pháp gọi phương thức trong ngữ cảnh của đối tượng là:
 - `objectname.methodname(parameters)`

Ví dụ Biểu thức quy tắc

Ví dụ:

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      re = /Time/
      str = re.test ("Time and Tide wait
        for none");
      window.alert(str);
    </SCRIPT>
  </HEAD>
</HTML>
```

Kết quả:





Mảng

- Mảng được dùng để lưu trữ một dãy các biến với cùng một tên.
- Một số (chỉ số) dùng để phân biệt các giá trị khác nhau.
- Các mảng bắt đầu với chỉ số 0 trong JavaScript.
- Tạo mảng: Cú pháp

```
arrayObjectName = new Array([element0, element1, ..., elementN])
```
- CỘNG các phần tử: Chúng ta có thể cộng các phần tử mảng khi chúng ta tạo nó. Ví dụ. `emp[0] = "Ryan Dias"`
- Các phần tử của một mảng có thể truy cập bằng tên **Name** hoặc chỉ số **Index** của phần tử.



Mảng(1)

- Các phương thức của đối tượng mảng có thể dùng thao tác trên mảng.
- Các phương thức của đối tượng mảng bao gồm:
 - join
 - pop
 - push
 - reverse
 - shift
 - sort
- JavaScript hỗ trợ mảng nhiều chiều.



Các câu lệnh điều kiện

- Câu lệnh điều kiện được dùng để kiểm tra điều kiện. Kết quả xác định câu lệnh hoặc khối lệnh được thực thi.
- Các câu lệnh điều kiện bao gồm:
 - If..... Else
 - Switch



LỆNH LẶP

- Cấu trúc điều khiển lặp trong chương trình là các lệnh lặp.
- Các kiểu lệnh lặp bao gồm:
 - For
 - Do While
 - While
 - Break & continue
 - For....in
 - with



Hàm

- JavaScript có sẵn các hàm định nghĩa trước dùng trong script.
- Một vài hàm định nghĩa trước trong JavaScript bao gồm:
 - Hàm eval
 - Hàm isNaN
- Hàm do người dùng tự tạo

```
function funcName(argument1, argument2, etc)
    { statements; }
```
- Gọi hàm
- Câu lệnh Return