Javascript là gì ?

- Ngôn ngữ thông dịch, mã nguồn của nó được nhúng hoặc tích hợp trực tiếp vào tập tin HTML. Khi trang web được tải xong, trình duyệt sẽ thông dịch và thực hiện các mã lệnh này.
- Được cung cấp hoàn toàn miễn phí

Javascript có thể làm gì?

- Làm cho trang HTML trở nên sinh động hơn.
- Phản ứng lại với một sự kiện nào đó từ phía người dùng.
- Đọc hoặc thay đổi nội dung của các phần tử trong trang HTML
- Kiểm tra dữ liệu
- Phát hiện các loại trình duyệt khác nhau
- Tạo các tập tin cookie lưu trữ và truy xuất thông tin trên máy tính của người truy cập website
- ...

Sử dụng Javascript như thế nào ?

- Để sử dụng Javascript rất đơn giản, chúng ta chỉ cần đưa các câu lệnh của nó vào trong thẻ <script> của HTML

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Example 01</title>
    <script type="text/javascript">
        document.write("Hello World!");
    </script>
</head>
<body>
</body>
</html>
```

Javascript thực thi lệnh khi nào ?

- TH1: Thực hiện lệnh ngay khi trang web được tải về trình duyệt của người sử dụng.
- TH2: Thực hiện lệnh khi nhận được một tác động nào đó như nhấn nút, di chuyển chuột, …

Vị trí Javascript trong trang HTML

- Đặt trong cặp thẻ <head> của trang web
- Đặt trong cặp thẻ <body> của trang web
- Đặt trong tập tin .js sau đó nhúng tập tin này vào trang web

Mã lệnh Javascript

- Mã lệnh javascript là một chuỗi các câu lệnh.
- Các câu lệnh này kết thúc bằng dấu chấm phẩy “;”
- Phân biệt chữ hoa và chữ thường
- Ký tự khoảng trắng không ảnh hưởng đến kết quả thực thi của mã lệnh.

Biến trong Javascript

```
x = 5
y = 6
z = x + y = 5 + 6 =11
```

- Biến dùng để lưu trữ một giá trị nào đó có thể là một chuỗi, một đối tượng, một con số, một mảng, một phép toán ...
- Khai báo một biến trong JavaScript: **var <tên_biến>;**

Quy tắc đặt tên biến

- Tên biến phải bắt đầu bằng ký tự hoặc dấu gạch dưới (_) và không có khoảng trắng.
- Tên biến phân biệt chữ hoa và chữ thường

Kiểu dữ liệu trong JavaScript

Kiểu	Ví dụ
String	`var answer = "It's alright";` `var answer = "He is called 'Johnny'";`
Number	`var x1 = 34.00;` `var x2 = 34;`
Boolean	`var x = true;` `var y = false;`

Array	`var cars=new Array();`
	`cars[0]="Saab";`
	`cars[1]="Volvo";`
	`cars[2]="BMW";`

Object	`var person={firstname:"John", lastname:"Doe", id:5566};`

Kiểm tra kiểu dữ liệu trong JavaScript

- Javascript là ngôn ngữ không ràng buộc về kiểu dữ liệu: không cần khai báo kiểu dữ liệu khi khai báo biến, một biến đang thuộc kiểu dữ liệu này có thể bị gán bởi một giá trị thuộc kiểu dữ liệu khác.
- Xác định kiểu dữ liệu của một biến ta sử dụng câu lệnh: typeof <tên biến>;

Lưu ý về kiểu dữ liệu của biến khi khai báo

```
<script type="text/javascript">
    var pi      = 3.14;
    var name    = "John Doe";
    var answer  = 'Yes I am!';
</script>
```

- Giá trị của một biến là một chuỗi khi và chỉ khi nó nằm trong cặp dấu ngoặc kép (“…”) hoặc cặp dấu ngoặc đơn (‘…’)
- Giá trị của một biến là một số khi và chỉ khi nó không nằm trong cặp dấu ngoặc kép (“…”) và không nằm trong cặp dấu ngoặc đơn (‘…’)

Sử dụng hàm trong JavaScript

Tại sao cần sử dụng hàm

- Xuất ra trình duyệt lời chào đối với mỗi thành viên trong diễn đàn ?

Sử dụng hàm trong JavaScript

Khai báo hàm

```
function function_name (var1, var2, … , varN){
    // code goes here
}
```

- var1, var2 … varN được gọi là các tham số của hàm. Hàm có thể có nhiều tham số hoặc không có tham sao nào cả
- Cách đặt tên hàm tương tự như cách đặt tên biến. Hoặc chúng ta dùng dấu gạch dưới (_) nếu tên hàm là một cụm từ.

Phân biệt biến cục bộ và biến toàn cục

Biến cục bộ (Local Variables)

- Phạm vi ảnh hưởng chỉ trong hàm mà nó được khai báo
- Vòng đời bắt đầu khi biến được khởi tạo
- Vòng đời kết thúc khi hàm thực hiện xong.

Biến toàn cục (Global Variables)

- Phạm vi ảnh hưởng đến toàn trang
- Vòng đời bắt đầu khi biến được khởi tạo
- Vòng đời kết thúc khi trang được đóng lại.

Toán tử trong JavaScript

Toán tử số học

Cho y = 9

Toán tử	Miêu tả	Ví dụ	Kết quả
+	Cộng	`x = y + 2`	`x = 11`
-	Trừ	`x = y - 2`	`x = 7`
*	Nhân	`x = y * 2`	`x = 18`
/	Chia	`x = y / 2`	`x = 4.5`
%	Lấy giá trị lẻ	`x = y % 2`	`x = 1`
++	Tăng	`x = ++y = y + 1`	`x = 10`
--	Giảm	`x = --y = y - 1`	`x = 8`

Toán tử trong JavaScript

Toán tử gán

Cho x = 10 và y = 5

Toán tử	Ví dụ	Hình thức khác	Kết quả
=	x = y		x = 5
+=	x += y	x = x + y	x = 15
-=	x -= y	x = x - y	x = 5
*=	x *= y	x = x * y	x = 50
/=	x /= y	x = x / y	x = 2
%=	x %= y	x = x % y	x = 0

Toán tử trong JavaScript

Toán tử so sánh

Cho x = 5

Toán tử	Mô tả	Ví dụ	Kết quả
==	So sánh bằng	x == 8	false
===	So sánh tuyệt đối	x === "5" x === 5	false true
!=	So sánh không bằng	x != 8	true
>	So sánh lớn hơn	x > 8	false
<	So sánh nhỏ hơn	x < 8	true
>=	So sánh lớn hơn hoặc bằng	x >= 8	false
<=	So sánh nhỏ hơn hoặc bằng	x <= 8	true

Toán tử trong JavaScript

Toán tử logic

Cho x = 6 và y = 3

Toán tử	Mô tả	Ví dụ	Kết quả
&&	And	(x < 10 && y > 1) (x < 10 && y > 4)	true false
||	Or	(x < 10 || y > 1) (x < 10 || y > 4)	true true
!	Not	!(x==y) !(x!=y)	true false

Toán tử trong JavaScript

Toán tử điều kiện

Cú pháp: `variablename = (condition) ? value1 : value2;`

```
<script type="text/javascript">
    var age = 5;
    var result = ( age > 5) ? "Vào cấp 1" : "Không được vào cấp 1";
    document.write(result);
</script>
```

Câu điều kiện trong Javascript

- Câu điều kiện là câu lệnh mà chúng ta thường xuyên sử dụng khi viết mã cho bất kỳ ngôn ngữ lập trình nào. Để thực hiện những hành động khác nhau trong những điều kiện khác nhau.
- Hai câu lệnh điều kiện thường được sử dụng trong JavaScript:
  - Câu điều kiện IF … ELSE
  - Câu điều kiện SWITCH

Câu điều kiện IF ... ELSE

Câu lệnh IF

- Sử dụng câu lệnh này để thực hiện một số mã lệnh nếu đúng điều kiện định ra.

```
<script type="text/javascript">
    if(condition) {
        // code goes here
    }
</script>
```

Câu điều kiện IF ... ELSE

Câu lệnh IF … ELSE

- Sử dụng câu lệnh điều kiện này để thực hiện một số mã lệnh nếu đúng điều kiện đã định và nếu không đúng điều kiện đã định thì thực hiện mã lệnh khác

```
<script type="text/javascript">
    if(condition) {
        // code goes here
    }else {
        // code goes here
    }
</script>
```

Câu điều kiện IF ... ELSE

Câu lệnh IF ... ELSE IF ... ELSE

- Sử dụng câu lệnh điều kiện này trong trường hợp có nhiều điều kiện đặt ra và khi thỏa mỗi điều kiện sẽ thực hiện một số mã lệnh khác nhau.

```
<script type="text/javascript">
    if(condition1) {
        // code goes here
    }else if(condition2){
        // code goes here
    }else {
        // code goes here
    }
</script>
```

Câu điều kiện SWITCH

- Câu điều kiện Switch có một điều kiện mặc định, nghĩa là khi giá trị đưa vào không thỏa một điều kiện nào thì nó sẽ lấy các câu lệnh trong phần điều kiện mặc định để thực hiện.

```
<script type="text/javascript">
    switch (expression) {
        case constant1:
            // execute code block 1
            break;
        case constant1:
            // execute code block 2
            break;
        default:
            // code to be executed if n is different from case 1 and 2
    }
</script>
```

Vòng lặp trong JavaScript

- Vòng lặp được dùng để thực thi một số việc nào đó cho đến khi đúng điều kiện thì thoát khỏi vòng lặp và thi hành lệnh tiếp theo
- Các vòng lặp thường được sử dụng trong JavaScript:
  - Vòng lặp FOR
  - Vòng lặp WHILE
  - Vòng lặp DO … WHILE
  - Vòng lặp FOR … IN

Vòng lặp FOR

```
<script type="text/javascript">
    for (startValue; endValue; varIncrement){
        //Code goes here
        /*
            startValue:     giá trị bắt đầu
            endValue:       giá trị cuối cùng
            varIncrement:   giá trị tăng của vòng lặp
        */
    }
</script>
```

Vòng lặp WHILE

- Vòng lặp while thực hiện một khối lệnh khi điều kiện thỏa và dừng lại ngay khi điều kiện không thỏa

```
<script type="text/javascript">
    while(expression){
        //Code goes here
    }
</script>
```

Vòng lặp DO ... WHILE

- Vòng lặp này sẽ thực hiện khối lệnh ít nhất một lần, rồi sau đó mới kiểm tra điều kiện. Khối lệnh vẫn sẽ được thực hiện khi biểu thức điều kiện vẫn còn đúng.

```
<script type="text/javascript">
    do {
        //Code goes here
    }while(expression);
</script>
```

Sử dụng break và continue trong vòng lặp

- Câu lệnh break có chức năng thoát khỏi một vòng lệnh. Nó có thể được sử dụng để nhảy ra khỏi một vòng lặp.
- Câu lệnh continue có chức năng dừng vòng lặp tại giá trị đó và nhảy sang giá trị khác trong vòng lặp

Ôn tập

```
0 1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9
2 3 4 5 6 7 8 9
3 4 5 6 7 8 9
4 5 6 7 8 9
5 6 7 8 9
6 7 8 9
7 8 9
8 9
9
```

```
          0
        1 0 1
      2 1 0 1 2
    3 2 1 0 1 2 3
  4 3 2 1 0 1 2 3 4
5 4 3 2 1 0 1 2 3 4 5
```

```
      6 5 4 3 2 1 0 1 2 3 4 5 6
    7 6 5 4 3 2 1 0 1 2 3 4 5 6 7
  8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8
9 8 7 6 5 4 3 2 1 0 1 2 3 4 5 6 7 8 9
```

JavaScript Errors - Throw and Try to Catch

- Các nguyên nhân gây ra lỗi khi thực thi JavaScript: lỗi cú pháp, lỗi đầu vào bị sai và các nguyên nhân khó xác định khác
- Trong lập trình có những ngoại lệ mà chúng ta ít để ý tới:
  - Phép chia giữa 2 số a và b, khi b bằng 0
  - Đọc và ghi file nhưng file chưa được tạo hay không có sẵn
  - Chưa điền dữ liệu vào text box nhưng vẫn đưa ra xử lý

JavaScript Errors - Throw and Try to Catch

- Cú pháp:

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

    try{
        //Run some code here
    }catch(err){
        //Handle errors here
    }

</script>
```