



JS1: Cơ bản về JS

FDP 5.0



Nội dung

- ❖ Giới thiệu JS
- ❖ Nhúng JS vào HTML
- ❖ Kiểu dữ liệu và cú pháp
- ❖ Hàm
- ❖ Cấu trúc điều kiện
- ❖ Vòng lặp
- ❖ Xử lý sự kiện

GIỚI THIỆU JS

❖ 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 JS

❖ **JavaScript** 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.

❖ Khi trình duyệt (*Client browser*) truy cập trang web có chứa các đoạn mã xử lý tại server-side. Server (run-time engine) sẽ thực hiện các lệnh *Server-side Scripts* và trả về nội dung HTML cho trình duyệt.

❖ Nội dung HTML trả về chủ yếu bao gồm: *mã html, client-script*.

NHÚNG JS VÀO HTML

❖ 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="abc.js"></script>
```

NHÚNG JS VÀO HTML

- ❖ Web Browser sẽ thực thi `<script>` khi load trang web theo thứ tự từ *trên xuống* dưới.
- ❖ Đối với *Source code JavaScript* có thể đặt trong các file `.js` sẽ được *nhúng* vào file HTML trước khi hoạt động.
- ❖ Các đoạn *code JavaScript* được Browser xử *cùng thứ tự* với các thẻ *HTML*. Trừ các *hàm (func/on)* chỉ được thực thi khi *có lời gọi hàm*.

NHÚNG JS VÀO HTML

- ❖ 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.

Kiểu dữ liệu và cú pháp

Biến

❖ 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 tên.

- ❖ Biến *không cần khai báo kiểu dữ* liệu vì biến trong javascript

- ❖ *không có kiểu dữ liệu nhất định*

Kiểu dữ liệu và cú pháp

Kiểu dữ liệu

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>connecton.Close();</code>	<code>connecton = null</code>

Kiểu dữ liệu và cú pháp

Đổi kiểu DL

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

❖ Ví dụ:

```
var x = 10;
```

```
x = "hello world !";
```

❖ 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.

Kiểu dữ liệu và cú pháp

Ví dụ

```
<!DOCTYPE html>
<html>
<body>
<h2>JS1</h2>
<p id="demo">Ví dụ </p>
<script>
  var x, y, z;
  x = 5;
  y = 6;
  z = x + y;
  document.getElementById("demo").innerHTML =
    "Tổng là: " + z + ".";
</script>
</body>
</html>
```

JS1

Tổng là: 11.

Câu lệnh hiển thị nội dung HTML lên tag có ID là "demo"

HÀM

❖ Dạng thức khai báo chung:

```
func/on Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
}
```

❖ Hàm có giá trị trả về:

```
func/on Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
    return (value);  
}
```

HÀM

Các quy 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* `*/`

HÀM

Ví dụ 1

```
<!DOCTYPE html>
<html>
<body>
<h2>Ví dụ 1 - Hàm không giá trị trả về</h2>
<HR>
<p id="demo"></p>

<script>
    function myFunction(p1, p2) {
        return p1 * p2;
    }
    document.getElementById("demo").innerHTML =
    "Tích = " + myFunction(4, 3);
</script>
</body>
</html>
```

Ví dụ 1 - Hàm không giá trị trả về

Tích = 12

HÀM

Ví dụ 2

```
<!DOCTYPE html>
<html>
<body>
<h2>Ví dụ 2 - Hàm có giá trị trả về</h2>
<HR>
<p id="demo"></p>

<script>
  var x = myFunction(4, 3);
  document.getElementById("demo").innerHTML =
    "Tích = " + x;

  function myFunction(a, b) {
    return a * b;
  }
</script>
</body>
</html>
```

Ví dụ 2 - Hàm có giá trị trả về

Tích = 12

CẤU TRÚC ĐIỀU KIỆN

Cấu trúc 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;
```

CẤU TRÚC ĐIỀU KIỆN

Switch case

switch (*expression*)

{

case *label1* :

statementlist

case *label2* :

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");
```

```
}
```

CẤU TRÚC ĐIỀU KIỆN

Ví dụ 1

```
<!DOCTYPE html>
<html>
<body>
<h1>Nhấn vào button hiển thị giờ và lời chào:</h1>
<button onclick="myFunction()">Nhấn vào đây</button>
<p id="demo"></p>
<script>
function myFunction() {
  var greeting;
  var time = new Date().getHours();
  if (time < 10) {
    greeting = "Good morning";
  } else if (time < 20) {
    greeting = "Good day";
  } else {
    greeting = "Good evening";
  }
  document.getElementById("demo").innerHTML =
  time + " giờ: " + greeting;
}
</script>
</body>
</html>
```

Nhấn vào button hiển thị giờ và lời chào:

Nhấn vào đây

14 giờ: Good day

CẤU TRÚC ĐIỀU KIỆN

Ví dụ 2

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
var day;
switch (new Date().getDay()) {
  case 0:
    day = "Chủ nhật";
    break;
  case 1:
    day = "Thứ 2";
    break;
  case 2:
    day = "Thứ 3";
    break;
  case 3:
    day = "Thứ 4";
    break;
```

```
  case 4:
    day = "Thứ 5";
    break;
  case 5:
    day = "Thứ 6";
    break;
  case 6:
    day = "Thứ 7";
}
document.getElementById("demo").innerHTML =
"Hôm nay là " + day;
</script>
</body>
</html>
```

Hôm nay là Thứ 4

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;
}
```

* for/in và break

VÒNG LẶP

While và do while

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

Ví dụ:

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

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

Ví dụ:

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

VÒNG LẶP

Ví dụ 1

```
<!DOCTYPE html>
<html>
<body>
<h2>Vòng lặp For</h2>
<p id="demo"></p>
<script>
    var text = "";
    var i;
    for (i = 0; i < 5; i++) {
        text += "Dòng: " + i + "<br>";
    }
    document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Vòng lặp For

Dòng: 0

Dòng: 1

Dòng: 2

Dòng: 3

Dòng: 4

VÒNG LẶP

Ví dụ 2

```
<!DOCTYPE html>
<html>
<body>
<h2>Vòng lặp While</h2>
<hr>
<p id="demo"></p>
<script>
  var text = "";
  var i = 0;
  while (i<5){
    text += "Dòng: " + i + "<br>";
    ++i;
  }
  document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Vòng lặp While

Dòng: 0
Dòng: 1
Dòng: 2
Dòng: 3
Dòng: 4

VÒNG LẶP

Ví dụ 3

```
<!DOCTYPE html>
<html>
<body>
<h2>Vòng lặp do while</h2>
<hr>
<p id="demo"></p>
<script>
  var text = "";
  var i = 0;
  do {
    text += "Dòng: " + i + "<br>";
    ++i;
  }while(i<5)
  document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

Vòng lặp do while

Dòng: 0
Dòng: 1
Dòng: 2
Dòng: 3
Dòng: 4

XỬ LÝ SỰ KIỆN

Sự kiện phổ biến

❖ 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 SỰ KIỆN

Xử lý sự kiện cho các thẻ HTML

❖ Cú pháp:

❖ **<TAG** **eventHandler** = "JavaScript Code">

❖ Lưu ý: Dấu “...” và ‘...’

```
<body>
  <input type="button"
         name="btnClickMe"
         value="Open new page"
         onclick="window.open('http://www.google.com');" />
</body>
```

XỬ LÝ SỰ KIỆN

Ví dụ

```
<!DOCTYPE html>
<html>
<body>

<h2>Hiển thị thời gian</h2>
<hr>
<button onclick="displayDate()">Nhấn vào đây</button>
<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>
<p id="demo"></p>
</body>
</html>
```

Hiển thị thời gian

Nhấn vào đây

Wed May 27 2020 15:30:28 GMT+0700 (Giờ Đông Dương)

Thank you and
happy learning !!!