

Phát triển ứng dụng WEB

Web Application Development

Giảng viên:

ThS. Hoàng Anh Đức

- Mobile: 0986999984 / 01222222638

- Email: ducha.hung@gmail.com

Bộ môn Công nghệ phần mềm, Phòng 703,
Nhà C12 tầng

GIỚI THIỆU

- JavaScript là một ngôn ngữ kịch bản, được thiết kế để tăng cường khả năng tương tác của trang HTML
- Là ngôn ngữ kịch bản thông dụng nhất trên Internet
- Là ngôn ngữ thông dịch
- Được hỗ trợ bởi tất cả các trình duyệt như Netscape, IE, Microsoft Edge, Cốc cốc, UC, .v.v..

JAVASCRIPT LÀM ĐƯỢC NHỮNG GÌ?

- Cung cấp cho những người thiết kế HTML một công cụ lập trình
- Cho phép đặt dữ liệu động vào trang HTML
- Cho phép tương tác với các sự kiện
- Cho phép đọc và ghi các phần tử HTML
- Cho phép kiểm tra tính hợp lệ của dữ liệu
- Cho phép tạo ra các cookie

INTERNAL JAVASCRIPT

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

...

```
</script>
```

Ví dụ :

```
<html><body>
```

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

```
    document.write("Hello World!")
```

```
</script>
```

```
</body> </html>
```

EXTERNAL JAVASCRIPT

```
<head>
```

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

```
</head>
```

BIẾN

- Sử dụng biến để lưu trữ thông tin
- Giá trị của biến có thể thay đổi được
- Cú pháp :

Var name = value;

Or

Name = value /*Cách thông dụng*/

PHẠM VI CỦA BIẾN

- Biến toàn cục : là biến được khai báo bên ngoài của một hàm, nó được tham chiếu ở mọi nơi trong tài liệu
- Biến cục bộ : là biến được khai báo bên trong một hàm và chỉ có thể sử dụng bên trong hàm đó.

PHÉP TOÁN

- Phép toán số học

<i>Operator</i>	<i>Used For</i>	<i>Example</i>	<i>Equals</i>
+	Addition	1+2	3
-	Subtraction	12-10	2
*	Multiplication	2*3	6
/	Division	10/3	3.3333333333
%	Modulus	10%3	1
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4
-	Unary negation	-20	negative 20

PHÉP TOÁN

- Phép gán:

<i>Operator</i>	<i>Example</i>	<i>Means</i>
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
%	$x\%=y$	$x=x\%y$

PHÉP TOÁN

- Phép so sánh và logic :

<i>Operator</i>	<i>Meaning</i>	<i>Example</i>
<code>==</code>	is equal to	<code>10==3</code> is false
<code>!=</code>	does not equal	<code>10!=3</code> is true
<code>></code>	is greater than	<code>10>3</code> is true
<code>>=</code>	is greater than or equal to	<code>10>=3</code> is true
<code><</code>	is less than	<code>10<3</code> is false
<code><=</code>	is less than or equal to	<code>10<=3</code> is false

<i>Operator</i>	<i>Meaning</i>	<i>If x=10 and y= 5 then...y</i>
<code>&&</code>	AND	<code>(x = 10) && (y < 10)</code> is true
<code> </code>	OR	<code>(x=10) (y=10)</code> is true
<code>!</code>	NOT	<code>x !=y</code> is true

CÁC 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 new
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>
function	<code>var add = new function("x", "y", "return(x+y)");</code> <code>add(2, 3);</code>	<i>functionName = new function([argname1, [... argnameN,]] body);</i>

ĐỔI KIỂU DỮ LIỆU

■ Chuyển chuỗi thành số

`parseInt();`

`parseInt("42")`

`parseInt("42.33")`

`parseFloat("42.33")`

`3 + 3 + parseInt("3")`

`parseFloat();`

`// result = 42`

`// result = 42`

`// result = 42.33`

`// result = 9`

■ Chuyển số thành chuỗi (`auto`):

`3 + "3"`

`3 + 3 + "3"`

`("" + 2500)`

`("" + 2500).length`

`// result = "33"`

`// result = "63"`

`// result = "2500"`

`// result = 4`

CÁC HẰNG SỐ TRONG MATH

Property	Description
<u>E</u>	Returns Euler's number (approx. 2.718)
<u>LN2</u>	Returns the natural logarithm of 2 (approx. 0.693)
<u>LN10</u>	Returns the natural logarithm of 10 (approx. 2.302)
<u>LOG2E</u>	Returns the base-2 logarithm of E (approx. 1.442)
<u>LOG10E</u>	Returns the base-10 logarithm of E (approx. 0.434)
<u>PI</u>	Returns PI (approx. 3.14)
<u>SQRT1_2</u>	Returns the square root of 1/2 (approx. 0.707)
<u>SQRT2</u>	Returns the square root of 2 (approx. 1.414)

Method	Description
<u>abs(x)</u>	Returns the absolute value of x
<u>acos(x)</u>	Returns the arccosine of x, in radians
<u>asin(x)</u>	Returns the arcsine of x, in radians
<u>atan(x)</u>	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
<u>atan2(y,x)</u>	Returns the arctangent of the quotient of its arguments
<u>ceil(x)</u>	Returns x, rounded upwards to the nearest integer
<u>cos(x)</u>	Returns the cosine of x (x is in radians)
<u>exp(x)</u>	Returns the value of E ^x
<u>floor(x)</u>	Returns x, rounded downwards to the nearest integer
<u>log(x)</u>	Returns the natural logarithm (base E) of x
<u>max(x,y,z,...,n)</u>	Returns the number with the highest value
<u>min(x,y,z,...,n)</u>	Returns the number with the lowest value
<u>pow(x,y)</u>	Returns the value of x to the power of y
<u>random()</u>	Returns a random number between 0 and 1
<u>round(x)</u>	Rounds x to the nearest integer
<u>sin(x)</u>	Returns the sine of x (x is in radians)
<u>sqrt(x)</u>	Returns the square root of x
<u>tan(x)</u>	Returns the tangent of an angle

VÍ DỤ SỬ DỤNG MATH

- `x = Math.random();`

`document.write(100*x);` //Hiển thị giá trị từ 0->100

- `a=6; b=10; c=9; d=8;`

`document.write("Số lớn nhất trong a,b,c,d là:" +
Math.max(a,b,c,d));`

CÁC THUỘC TÍNH CỦA STRING

- length: Chiều dài chuỗi
- constructor: Dùng để kiểm tra kiểu của biến
- +: Dùng để kết nối các chuỗi
- ...

Ví dụ:

```
Str = "Hello world!";
```

```
document.write(Str.length); //Hiển thị số 12
```

```
document.write(Str.constructor); //Hiển thị "String"
```


CÁC PHƯƠNG THỨC CỦA STRING

- **anchor**
- **big**
- **blink**
- **bold**
- **charAt**
- **charCodeAt**
- **concat**
- **fixed**
- **fontcolor**
- **fontsize**
- **fromCharCode**
- **indexOf**
- **italics**
- **lastIndexOf**
- **link**
- **localeCompare**
- **match**
- **replace**
- **search**
- **slice**
- **small**
- **split**
- **strike**
- **sub**
- **substr**
- **substring**
- **sup**
- **toLocaleLowerCase**
- **toLocaleUpperCase**
- **toLowerCase**
- **toUpperCase**
- **toString**
- **valueOf**

CÁC PHƯƠNG THỨC CỦA STRING

Phương thức	Ví dụ
strObj.charAt(index)	<pre>var str = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"; var s1 = str.charAt(1); // s1 = "B" var s2 = str.charAt(26); // s1 = null</pre>
strObj.charCodeAt(index)	<pre>var n = str.charCodeAt(1); //Get the Unicode char at pos 1.</pre>
strObj.concat([string2[, string3[,... [, stringn]]])	<pre>var str1 = "ABCDEFGHJKLM" var str2 = "NOPQRSTUVWXYZ"; var s = str1.concat(str2); // s = "ABC...NOP...XYZ"</pre>
strObj.fromCharCode([cod e1[, code2[,... [, codeN]]])	<p>Tạo chuỗi từ mã ngoài Unicode.</p> <pre>var test = String.fromCharCode(112, 108, 97, 105, 110);</pre>
strObj.indexOf(subString[, startIndex])	<pre>var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.indexOf("BEB", 0); // s = 2</pre>
strObj.lastIndexOf(substring[, startindex])	<pre>var str1 = "BABEBIBOBUBABEBIBOBU" var s = str1.lastIndexOf("BEB", 0); // s = 12</pre>

CÁC PHƯƠNG THỨC CỦA STRING

Phương thức	Ví dụ
strObj.match(rgExp) rgExp = /pattern/[flags] flags, may be combined: g (global search for all occurrences of pattern) i (ignore case) m (multiline search)	<pre>var s = "The rain in Spain falls mainly in the plain"; var re = /The/gi; //Create regular expression pattern. var r = s.match(re); // r = {"The", "the"}.</pre>
strObj.replace(rgExp, replaceText)	Tìm và thay thế.
stringObj.search(rgExp)	Có xuất hiện, cho ra vị trí. Ngược lại -1.
stringObj.slice(start, [end])	Lấy chuỗi con.
stringObj.split([separator[, limit]]) separator: string or Regular Expression	<pre>var s = "The rain in Spain falls mainly in the plain."; var ss = s.split(" ", 3); // ss = {"The"; "rain"; "in"}</pre>
string.anchor(anchorString)	<pre>var string = "Information about Fish"; var str_anchor = string.anchor("fish_infor"); window.document.writeln(str_anchor); // < A name="fish_infor">Information about fish</pre>

Kiểu Number

- Kiểu dữ liệu số nguyên, số thực
- Hệ thập phân
- Hệ nhị phân
- Hệ bát phân
- Hệ thập lục phân
- Ví dụ: Cho biết giá trị tương ứng:
125 => bát phân, thập lục phân, nhị phân
34F4 => thập phân, bát phân, nhị phân

Thuộc tính của Number

- Constructor
- Prototype
- MAX_VALUE (khoảng = $1.79E+308$)
- MIN_VALUE (khoảng = $5.00E-324$)
- NaN (Not a Number)
- NEGATIVE_INFINITY ($= -MAX_VALUE$)
- POSITIVE_INFINITY ($= -MIN_VALUE$)

Phương thức của Number

- toExponential

numObj.**toExponential**(*[fractionDigits]*)

- toFixed

numObj.**toFixed**(*[fractionDigits]*)

- toPrecision

numObj.**toPrecision**(*[precision]*)

- toString *objectname*.**toString**(*[radix]*)

radix : [2; 8; 10; 16]

Hàm Date()

- `var dateObj = new Date()`
- `var dateObj = new Date(dateVal)`
- `var dateObj = new Date(year, month, date[, hours[,minutes[,seconds[,ms]]]])`
- `var dateObj = new Date(dateString)`
- `var dateObj = new Date(milliseconds) //Tính từ 0h ngày 1/1/1970`

Phương thức trong Date

- getDate
- getDay
- getFullYear
- getHours
- getMilliseconds
- getMinutes
- getMonth
- getSeconds
- getTime
- getTimezoneOffset
- getVarDate
- getYear
- setDate
- setFullYear
- setHours
- setMilliseconds
- setMinutes
- setMonth
- setSeconds
- setTime
- setYear
- parse
- toString
- toDateString
- toGMTString
- toLocaleDateString
- toLocaleString
- toLocaleTimeString
- toTimeString
- toUTCString

Mảng (Array)

Khai báo:

```
var arrayObj = new Array()
```

```
var arr = []
```

```
var arrayObj = new Array([size])
```

```
var arrayObj = new Array(element0,element1...elementn)
```

```
Var arr = new Array(start_element, end_element)
```

```
var arrayObj = new Array(Array)//Không sử dụng  
được nữa
```

Chú ý: chỉ số Array bắt đầu bằng 0

Thuộc tính của Array

- length
- constructor
- prototype
- ...

Phương thức của Array

Phương thức	Ví dụ
array1.concat ([item1[, item2[,...,...,... [, itemN]]]])	<pre>var a, b, c, d; a = new Array(1,2,3); b = "JScript"; c = new Array(42, "VBScript"); d = a.concat(b, c); //Returns the array [1, 2, 3, "JScript", 42, "VBScript"]</pre>
arrayObj.join (separator)	<pre>var a, b; a = new Array(0,1,2,3,4); b = a.join("-"); // b = "0-1-2-3-4"</pre>
arrayObj.pop ()	<pre>var n = a.pop(); // n = 4</pre>
arrayObj.push ([item1 [item2 [. . . [itemN]]])	<pre>a.push(5); // a = [0,1,2,3,5]</pre>
arrayObj.reverse ()	<pre>a.reverse(); // a = [5,3,2,1,0]</pre>
arrayobj.sort (sortFunction)	<pre>function numerical_sorter(n1,n2) { return (n1 - n2); } var l = a.sort(); // l = [0,1,2,3,5].</pre>

Array nhiều chiều

- Không còn hỗ trợ, chỉ khi khai báo từng phần tử của 1 mảng là 1 mảng khác.

CÁC LỆNH ĐIỀU KHIỂN

- Câu lệnh if
if (*condition*)

{

...

}

- Câu lệnh if ... else
if (*condition*)

{

...

} else

{

...

}

CÂU LỆNH IF – VÍ DỤ

```
<script type="text/javascript">  
    d=new Date()  
    var time=d.getHours()  
    if (time<10)  
    {  
        document.write("<b>Good morning</b>")  
    }  
</script>
```

SWITCH

```
switch(n)
{
    case 1:
        ...
        break
    case 2:
        ...
        break
    default:
        ...
}
```

SWITCH – VÍ DỤ

```
<script type="text/javascript">
var d=new Date();
theDay=d.getDay();
switch (theDay) {
  case 5:
    document.write("Finally Friday");
    break;
  case 6:
    document.write("Super Saturday");
    break ;
  case 0:
    document.write("Sleepy Sunday");
    break ;
  default: document.write("I'm looking forward to this weekend!");}
</script>
```


FOR

```
for(var=start;var<=end;var=var+increment)
{
    ....
}
```

FOR – VÍ DỤ

```
<script type="text/javascript">  
    //var i=0  
    for (i=0;i<=10000;i+=2)  
    {  
        document.write("The number is " + i)  
        document.write("<br />")  
    }  
</script>
```

WHILE

```
while (conditions)
```

```
{
```

```
...
```

```
}
```

Hoặc

```
do {
```

```
...
```

```
}
```

```
while (conditions)
```

WHILE – VÍ DỤ

```
i=0;
```

```
while(i<=10)
```

```
{
```

```
    document.write("The number is " + i);
```

```
    document.write("<br />");
```

```
    i=i+1;
```

```
}
```

WHILE – VÍ DỤ

```
i=0;
```

```
do{
```

```
    document.write("The number is " + i);
```

```
    document.write("<br />");
```

```
    i=i+1;
```

```
} while(i<=10)
```

POPUP BOXES

- Alert Box :
 - Cú pháp : `alert("sometext")`
 - Ví dụ :

```
<body>  
  <script type="text/javascript">  
    alert("Hello World");  
  </script>  
</body>
```

POPUP BOXES

- Confirm Box :

- Cú pháp

- `var r = confirm("sometext")`

- Ví dụ :

- `<Script type="text/javascript">`

- `var r=confirm("Press a button")`

- `if (r==true){`

- `document.write("You pressed OK!")`

- `}`

- `else{`

- `document.write("You pressed Cancel!")`

- `}`

- `</Script>`

POPUP BOXES

- Prompt Box
 - Cú pháp : `prompt("sometext","default value")`
 - Ví dụ :

```
<script type="text/javascript">  var
    r=prompt("Test","type here")
    document.write(r)
</script>
```


FUNCTIONS

- Cú pháp

- Không tham số :

```
function functionname()
```

```
{ some code }
```

- Có tham số

```
function functionname(var1, var2, ..., varX)
```

```
{ some code }
```

FUNCTIONS – VÍ DỤ

```
function minus(x,y){  
    return x-y;  
}
```

//Gọi hàm :

```
a = minus(3456,8765);  
document.write(a);
```

EVENTS

- Sử dụng JavaScript để đáp lại các sự kiện phát sinh trên trang HTML
- Các sự kiện có thể là :
 - Click chuột
 - Trang Web hay ảnh được tải
 - Lướt chuột trên trang Web
 - Submit 1 form
 - Một phím trên bàn phím được bấm
 - ...vv

HTML EVENTS

- OnLoad
- OnClick
- OnDbIClick
- OnMouseMove
- OnFocus
- OnSubmit
- OnChange
- OnInput

And more... (Tham khảo thêm w3schools.com)