

Chương 3

JAVASCRIPT

Nguyen Thai Khanh Quyen

1

Giới thiệu

- Javascript là ngôn ngữ kịch bản dựa trên đối tượng, thường được dùng để viết các ứng dụng Internet
- Các thao tác thường sử dụng javascript:
 - Tương tác với người
 - Thay đổi nội dung động
 - Kiểm tra tính hợp lệ dữ liệu

2

Chèn js vào HTML

- Chèn trực tiếp vào <body> hoặc <head> bằng thẻ <script>...</script>

- Ví dụ:

```
<head>
  <script>
    function myFunction()
    {
      document.getElementById("demo").innerHTML
="My First JavaScript Function";
    }
  </script>
</head>
```

3

Chèn js vào HTML

- Liên kết ngoài (External JavaScripts)

- Ví dụ:

```
<body>
  <script src="myScript.js"></script>
</body>
```

4

Khai báo biến

- Tên biến bắt đầu bằng chữ cái hoặc dấu _
- Phân biệt hoa thường
- Khai báo:
 - Var x=4;
 - X=4;
 - Var a,b,c;
- Biến toàn cục: không cần từ khóa var
- Biến cục bộ: phải sử dụng từ khóa var khi khai báo

5

Các kiểu dữ liệu

- Các kiểu dữ liệu:
 - String: chuỗi
 - Number: số nguyên, số thực
 - Boolean: true, false
 - Array: mảng
 - Object: đối tượng
 - Null: giá trị null
- Khi khai báo không cần chỉ rõ kiểu dữ liệu của biến, kiểu dữ liệu sẽ tùy vào giá trị được gán cho biến đó

6

Các kiểu dữ liệu

- Ví dụ:
- `var x; // undefined`
- `var x = 5; // number`
- `var x = "John"; // string`
- `var x=true; //boolean`
- `var x=["Saab","Volvo","BMW"]; //array`
- `var x={firstname : "John", lastname : "Doe", id : 5566 }; //object`
- `x=null; //null`

7

Các kiểu dữ liệu

- Ví dụ:
- `x = "hello javascript" + 5; → ???`
- `x = "10" + 5.6; → ???`
- `x= parseInt("10") + 5.6; → ???`
- `x=parseFloat("12.5") + 5.6; → ???`
- `x=parseFloat("a12.5") + 5.6; → ???`

8

Toán tử

Toán tử số học

Operator	Example
+	$x=y+2$
-	$x=y-2$
*	$x=y*2$
/	$x=y/2$
%	$x=y\%2$
++	$x=++y$
	$x=y++$
--	$x=--y$
	$x=y--$

Toán tử

• Toán tử so sánh

Operator	Example	Same As
=	$x=y$	
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
%=	$x\%=y$	$x=x\%y$

Toán tử

- Toán tử logic

Operator	Example	Description
And (&&)	expr1 && expr2	Trả về giá trị của expr1 nếu nó là false, nếu không thì trả về giá trị của expr2.
Or ()	expr1 expr2	Trả về giá trị của expr1 nếu nó là true, nếu không thì trả về giá trị của expr2.
Not (!)	!expr	Trả về giá trị false nếu biểu thức đúng và trả về giá trị true nếu biểu thức sai.

11

Câu lệnh điều kiện

- Câu lệnh if
 - if (*condition*)
 - { *code to be executed if condition is true* }
 - if (*condition*)
 - { *code to be executed if condition is true* }
 - else
 - { *code to be executed if condition is not true* }

12

Câu lệnh điều kiện

- Câu lệnh switch
 - switch(n)
 - {
 - case 1:
 - execute code block 1*
 - break;
 - ...
 - case m:
 - execute code block 2*
 - break;
 - default:
 - code to be executed if n is different from case 1 to case m*
 - }

13

Vòng lặp

- Vòng lặp for:
 - for (statement 1; statement 2; statement 3)
 - {
 - the code block to be executed*
 - }
 - statement 1 : khởi tạo biến đếm
 - statement 2: điều kiện của vòng lặp.
 - statement 3: thay đổi biến đếm trong vòng lặp.

Ví dụ:

```
for (var i=0;i<5;i++)
{
  x="The number is " + i + "<br>";
}
```

14

Vòng lặp

- Vòng lặp for in (dùng cho object)

Ví dụ

```
var txt="";
var person={fname:"John",lname:"Doe",age:25};
for (var x in person)
{
    txt=txt + person[x];
}
```

15

Vòng lặp

- Vòng lặp while


```
while (condition)
{
    code block to be executed
}
```

Ví dụ:

```
while (i<5)
{
    x=x + "The number is " + i + "<br>";
    i++;
}
```

16

Vòng lặp

- do/while

```
do
{
    code block to be executed
}
while (condition);
```

Ví dụ:

```
do
{
    x=x + "The number is " + i + "<br>";
    i++;
}
while (i<5);
```

17

Vòng lặp

- Lệnh break và continue
 - break: dừng vòng lặp
 - continue: dừng lần lặp hiện tại và quay lại kiểm tra điều kiện để thực hiện lần lặp tiếp theo.

18

Hàm

- Khai báo

```
function functionname()
{
    some code to be executed
}
```
- Khai báo hàm có đối số

```
function myFunction(var1,var2)
{
    some code
}
```
- Khai báo hàm có giá trị trả về

```
function myFunction()
{
    Some code
    return return_value;
}
```

19

Các đối tượng cơ bản (object)

- Hầu hết mọi thành phần trong javascript đều có thể là một đối tượng (object)
- Đối tượng là dữ liệu (biến), có thuộc tính (property) và phương thức (method)
- Ví dụ:

Object	Property	Method
"Hello World"	txt.length	txt.indexOf("World")

20

Các đối tượng cơ bản (object)

- **Tạo object:**

- Cách 1: trực tiếp

```
var txt = new String("Hello World");  
Var person=new Object();  
    person.firstname="Tran";  
    person.lastname="Hong";  
    person.age=30;  
person={firstname:"Tran",lastname:"Hong",age:30};
```

21

Các đối tượng cơ bản (object)

- **Cách 2: dùng hàm tạo đối tượng**

```
function person(firstname,lastname,age)  
{  
    this.firstname=firstname;  
    this.lastname=lastname;  
    this.age=age;  
}  
myFriend=new person("John","Doe",50);
```

22

Các đối tượng cơ bản (object)

- Truy cập thuộc tính của object
 - *objectName.propertyName*
 - Ví dụ: *person.firstname.length*;
- Truy cập phương thức của object
 - *objectName.methodName()*
 - ví dụ: *person.firstname.toUpperCase()*;

23

Các đối tượng cơ bản (object)

Phân cấp các đối tượng



Ví dụ

- Truy cập đối tượng form: **document.form**
- Truy cập đối tượng văn bản text1 trong form:
document.form.text1.value

24

Các đối tượng cơ bản (object)

Đối tượng string

- Đối tượng string được dùng để thao tác và làm việc với chuỗi văn bản
- Thuộc tính

Property	Description
length	Chiều dài chuỗi Ví dụ: <code>var str="Hello World!";</code> <code>var n=str.length;</code>

25

Các đối tượng cơ bản (object)

Đối tượng string

- Các phương thức

Method	Syntax	Description
charAt()	<code>string.charAt(index)</code>	Trả về kí tự tại vị trí n. <code>var str = "HELLO WORLD";</code> <code>var res = str.charAt(0) ;</code>
concat()	<code>string.concat(string 1, ...string n)</code>	Nối chuỗi <code>var str1 = "Hello "; var str2 = "world!";</code> <code>var res = str1.concat(str2);</code>
indexOf()	<code>string.indexOf(searchvalue,[start])</code>	Trả về vị trí của kí tự đầu tiên của chuỗi con trong chuỗi mẹ <code>var str = "Hello world, welcome to the universe.";</code> <code>var n = str.indexOf("welcome");</code>

26

Các đối tượng cơ bản (object) Đối tượng string

- Các phương thức

Method	Syntax	Description
lastIndexOf()	<i>string.lastIndexOf(searchvalue,[start])</i>	Trả về vị trí cuối cùng của chuỗi con trong chuỗi mẹ var str = "Hello planet earth, you are a great planet."; var n = str.lastIndexOf("planet");
replace()	<i>string.replace(searchvalue,newvalue)</i>	Thay thế chuỗi Ví dụ: var str = "Hello Microsoft!"; var res = str.replace("Microsoft","Google");
search()	<i>string.search(searchvalue)</i>	Trả về vị trí được tìm thấy của chuỗi con trong chuỗi mẹ Ví dụ: var str = "hello my friend"; var n = str.search("friend");

27

Các đối tượng cơ bản (object) Đối tượng string

- Các phương thức

Method	Syntax	Description
substr()	<i>string.substr(start,length)</i>	Tách chuỗi tại vị trí start với chiều dài length Ví dụ: var str = "Hello world!"; var res = str.substr(1,4)
substring()	<i>string.substring(start,[end])</i>	Tách chuỗi tại vị trí start đến vị trí end
toString()	<i>string.toString()</i>	
toLowerCase()	<i>string.toLowerCase()</i>	Chuyển thành chữ thường
toUpperCase()	<i>string.toUpperCase()</i>	Chuyển thành chữ hoa

28

Các đối tượng cơ bản (object) Đối tượng string

- Ví dụ:**

```
s1 = new String("HELLO");
s2 = new String ("world");
s = s1 + " " + s2;
n = s.indexOf("wo");
document.write("s1 in lower case: " + s1.toLowerCase());
document.write("s2 in upper case: " + s2.toUpperCase());
document.write("Charater at 4th position in s2 is: "+
s2.charAt(3));
document.write("The position of s2 in s: " + s.indexOf(s2));
document.write("The length of s: " + s.length);
document.write("Extract 5 characters from s at 4th position: "
+ s.substr(4,5));
```

29

Các đối tượng cơ bản (object) Đối tượng Math

Thuộc tính

Property	Description
E	Giá trị số Eurler (≈ 2.718)
LN2	Logarit cơ số 2 (≈ 0.693)
LN10	Logarit cơ số 10 (≈ 2.302)
LOG2E	Logarit cơ số 2 của E (≈ 1.442)
LOG10E	Logarit cơ số 10 của E (≈ 0.434)
PI	PI (≈ 3.14)
SQRT1_2	Căn bậc 2 của 1/2 (≈ 0.707)
SQRT2	Căn bậc 2 của 2 (≈ 1.414)

Ví dụ: `n = Math.PI;`

30

Các đối tượng cơ bản (object) Đối tượng Math

- Phương thức

Method	Description
Abs (number)	Trả về giá trị tuyệt đối của một số
Sin (number)	Trả về giá trị sin của của một số (tính bằng radian)
Cos (number)	Trả về giá trị cosin của của một số (tính bằng radian)
Tan (number)	Trả về giá trị tang của của một số (tính bằng radian)

Ví dụ: `n = Math.sin(45);`

31

Các đối tượng cơ bản (object) Đối tượng Math

- Phương thức

Method	Description
Min (number1, number2)	Trả về giá trị nhỏ nhất của hai số number1 và number2
Max (number1, number2)	Trả về giá trị lớn nhất của hai số number1 và number2
Round (number)	Làm tròn đối số tới số nguyên gần nhất
Sqrt (number)	Trả về căn bậc hai của một số

32

Các đối tượng cơ bản (object) Đối tượng Date

- Phương thức

Method	Description
getDate()	Trả về ngày của tháng(from 1-31)
getDay()	Trả về ngày của tuần (from 0-6)
getFullYear()	Năm (four digits)
getHours()	Giờ (from 0-23)
getMilliseconds()	Mili giây(from 0-999)
getMinutes()	Phút(from 0-59)
getMonth()	Tháng (from 0-11)
getSeconds()	Giây (from 0-59)
getTime()	Trả về số giây tính từ ngày 1/1/1970

33

Các đối tượng cơ bản (object) Đối tượng Date

- Phương thức

Method	Description
setDate()	Thiết lập ngày trong tháng cho đối tượng
setFullYear()	Thiết lập giá trị năm cho đối tượng
setHours()	Thiết lập giờ
setMilliseconds()	Thiết lập mili giây
setMinutes()	Thiết lập phút
setMonth()	Thiết lập tháng
setSeconds()	Thiết lập giây
setTime()	Thiết lập thời gian (tính bằng mili giây)

34

Các đối tượng cơ bản (object) Đối tượng Date

- **Ví dụ:**

```
function date()
```

```
{
```

```
    mydate = new Date();
```

```
    d = mydate.getDay();
```

```
    da = mydate.getDate();
```

```
    m = mydate.getMonth();
```

```
    y = mydate.getFullYear();
```

```
    sd = "";
```

```
    switch(d)
```

```
    {
```

```
        case 0:
```

```
            sd = "Sunday";
```

```
            break;
```

```
        case 1:
```

```
            sd = "Monday";
```

```
            break;
```

```
        case 2:
```

```
            sd = "Tuesday";
```

```
            break;
```

35

Các đối tượng cơ bản (object) Đối tượng Date

```
        case 3:
```

```
            sd = "Wednesday";
```

```
            break;
```

```
        case 4:
```

```
            sd = "Thursday";
```

```
            break;
```

```
        case 5:
```

```
            sd = "Friday";
```

```
            break;
```

```
        case 6:
```

```
            sd = "Saturday";
```

```
            break;
```

```
        default:
```

```
            break;
```

```
    }
```

```
    document.title = "Today is
```

```
        " + sd;
```

36

Các đối tượng cơ bản (object)

Đối tượng Array

Thuộc tính

Property	Description
length	Số phần tử của mảng

Phương thức

Method	Syntax	Description
concat()	<i>array1.concat(array2, array3,..., arrayX)</i>	Nối mảng
indexOf()	<i>array.indexOf(item,[start])</i>	Tìm vị trí của phần tử item trong mảng
join()	<i>array.join([separator])</i>	Nối các phần tử trong mảng để tạo
lastIndexOf()	<i>array.lastIndexOf(item,[start])</i>	Trả về vị trí cuối cùng của phần tử item

37

Các đối tượng cơ bản (object)

Đối tượng Array

Method	Syntax	Description
pop()	<i>array.pop()</i>	Xóa phần tử cuối mảng và trả về giá trị đó
push()	<i>array.push(item1, item2, ..., itemX)</i>	Thêm phần tử vào cuối mảng
reverse()	<i>array.reverse()</i>	Đảo ngược thứ tự mảng
shift()	<i>array.shift()</i>	Xóa phần tử đầu mảng và trả về giá trị đó
slice()	<i>array.slice(start,[end])</i>	Tách mảng từ vị trí start
sort()	<i>array.sort()</i>	Sắp xếp mảng

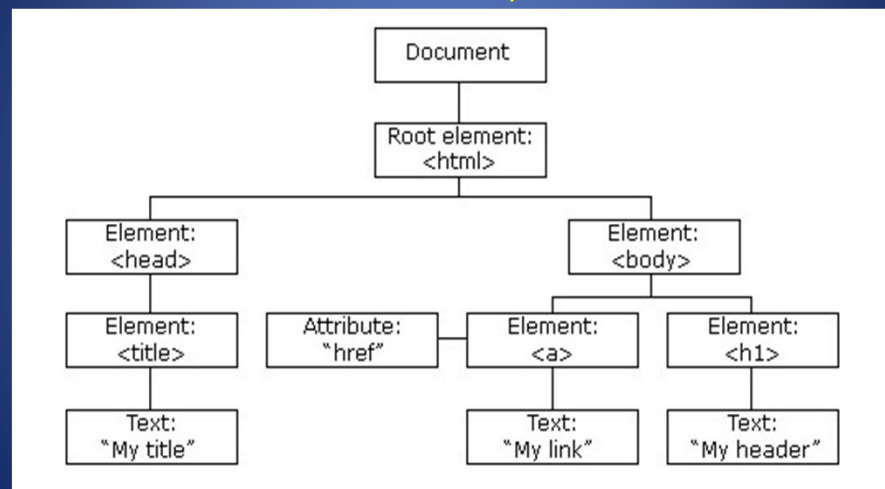
38

HTML DOM (Document Object Model)

- HTML Dom cung cấp một chuẩn cho việc tác động lên các thành phần HTML như: lấy giá trị, thay đổi, thêm, xóa,... Nó định nghĩa:
 - Các thành phần HTML được xem như là các objects
 - Các thuộc tính của các thành phần HTML
 - Các phương thức truy cập thành phần HTML
 - Các sự kiện

39

HTML DOM (Document Object Model)



40

HTML DOM Document

Method	Description
<code>document.getElementById()</code>	Tìm thành phần HTML theo tên id (Đây là cách dễ nhất và thông dụng nhất)
<code>element.innerHTML=</code>	Thay đổi nội dung thành phần HTML
<code>element.attribute=</code>	Thay đổi thuộc tính của thành phần HTML
<code>element.style.property=</code>	Thay đổi kiểu của thành phần HTML

Ví dụ 1:

```
<body>
  <h1 id="demo1" align="center">Hello world</h1>
  <p>This is a paragraph 1</p>
  <p>This is a paragraph 2</p>
  <p>This is a paragraph 3</p>
```

41

HTML DOM Document

```
• <script>
  x = document.getElementById("demo1");
  y = document.getElementsByTagName("p");
  x.innerHTML = "Hello world again";
  x.align = "right";
  y[1].setAttribute("align","center");
  x.style.color = "red";
  y[1].style.backgroundColor = "blue";
  document.write(x.innerHTML);
  document.write(y[1].innerHTML);
</script>
</body>
```

42

HTML DOM Document

- Ví dụ 2:

```
<body>
  <form id="frm1" action="form_action.asp">
    First name: <input type="text" name="fname"
      value="ABC"><br>
    Last name: <input type="text" name="lname"
      value="DEF"><br>
    <input type="submit" value="Submit">
  </form>
  <p id="demo"></p>
  <button onclick="clickIt()">Try it</button>
```

43

HTML DOM Document

```
<script>
  function clickIt()
  {
    var x = document.getElementById("frm1");
    var t = "";
    for (var i=0;i<x.length;i++)
    {
      x.elements[i].style.color = "red";
      t = t + x.elements[i].value + "<br>";
    }
    document.getElementById("demo").innerHTML=t;
  }
</script>
```

44

HTML DOM Events

Property	Description
Mouse Events	
onclick	Sự kiện click chuột
ondblclick	Sự kiện double click chuột
onmousedown	Sự kiện khi chuột được nhấn
onmousemove	Sự kiện di chuyển chuột trong 1 thành phần
onmouseover	Sự kiện rê chuột lên 1 thành phần
onmouseout	Sự kiện di chuyển chuột ra khỏi 1 thành phần
onmouseup	Sự kiện nhả chuột sau khi được nhấn xuống

45

HTML DOM Events

Attribute	Description
Keyboard Events	
onkeydown	Sự kiện khi người dùng đang nhấn phím
onkeypress ⁽¹⁾	Sự kiện khi người dùng nhấn phím
onkeyup	Sự kiện khi người dùng nhả phím
Frame/Object Events	
onload	Sự kiện xảy ra khi document, frameset hoặc object được load xuống
onresize	Sự kiện xảy ra khi thay đổi kích cỡ tài liệu
onscroll	Sự kiện xảy ra khi frame/document được cuộn
onunload	Sự kiện khi ngừng load trang web (for <body> and <frameset>)

46

HTML DOM Events

Attribute	Description
Form Events	
onchange	Sự kiện xảy ra khi có sự thay đổi trong các thành phần của form (for <input>, <select>, and <textarea>)
onfocus	Sự kiện xảy ra khi một phần tử của form trở thành phần tử hiện thời (for <label>, <input>, <select>, <textarea>, and <button>)
onblur	Ngược lại với onfocus
onreset	Sự kiện khi form được reset
onselect	Sự kiện khi người dùng chọn text (for <input> and <textarea>)
onsubmit	Sự kiện khi form được submit

47

HTML DOM Events

- Ví dụ 3:

```

<script>
function bigImage(i)
{
    i.style.width = "600px";
    i.style.height = "402px";
}
function normal(i)
{
    i.style.width = "300px";
    i.style.height = "201px";
}

```

48

HTML DOM Events

Ví dụ 3(tt):

```
function changebgcolor()
{
    x = document.getElementById("fname");
    x.style.backgroundColor = "yellow";
}
function uppertext()
{
    x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}
</script>
```

49

HTML DOM Events

Ví dụ 3(tt)

```
<body>
  <h1 onclick="this.innerHTML='You have just clicked this
  text!'">Click on this text!</h1>
  <h1 ondblclick="this.innerHTML='You have just double
  clicked this text!'">Double click on this text!</h1>
  
  <p>Enter your name: <input type="text" id="fname"
  onfocus="changebgcolor();" onblur="uppertext()"></p>
</body>
```

50

Browser Object Model (BOM)

- BOM cho phép javascript “giao tiếp” với browser
- Không có một chuẩn chung cho BOM
- Các đối tượng thường sử dụng:
 - Window
 - Screen
 - Location
 - History
 -

51

Browser Object Model (BOM) Window Object

Thuộc tính

Property	Description
document	Trả về đối tượng document của window
event	Thể hiện trạng thái của một sự kiện
history	Lưu trữ thông tin của những URL đã được viếng thăm.
location	Lưu trữ thông tin của URL hiện tại.
name	Thiết lập hoặc khôi phục tên của window hoặc Frame.
navigator	Lưu trữ thông tin của trình duyệt.
screen	Lưu trữ thông tin về kích cỡ và dạng đầy đủ của màn hình mà người dùng sử dụng.
innerHeight	Độ cao của vùng nội dung cửa sổ
innerWidth	Độ rộng của vùng nội dung cửa sổ
length	Trả về số lượng các frame/iframe trong cửa sổ

52

Browser Object Model (BOM)

Window Object

- Phương thức

Method	Description
Alert	Hiển thị một hộp chứa thông báo do ứng dụng định nghĩa.
Blur	Làm cho đối tượng mất focus và kích hoạt sự kiện onblur.
Close	Đóng cửa sổ trình duyệt hiện thời hoặc ứng dụng HTML.
Focus	Đặt một điều khiển là focus và thực thi đoạn lệnh được chỉ ra trong sự kiện onload.
navigate	Tải URL được chỉ định vào cửa sổ hiện thời.
open	Mở cửa sổ mới và tải tài liệu được chỉ ra bởi một URL hoặc mở một tài liệu trống nếu không có một URL nào được chỉ định.

53

Browser Object Model (BOM)

Window Object

- Ví dụ 4:

```
<script>
```

```
var w=window.innerWidth ||
document.documentElement.clientWidth ||
document.body.clientWidth;
var h=window.innerHeight ||
document.documentElement.clientHeight ||
document.body.clientHeight;
x=document.getElementById("demo");
x.innerHTML="Browser inner window width: " + w + ",
height: " + h + ".";
var newWin;
```

54

Browser Object Model (BOM) Window Object

Ví dụ 4(tt)

```
function openwindow()
{
    newWin =
    window.open("date.html","newWin","width=800,height=600");
}
function closeWindow()
{
    newWin.close();
}
</script>
<p id="demo"></p>
<p>Click the button to open a new browser window.</p>
<button onclick="openwindow()">Open new window</button>
<button onclick="closeWindow()">Close new window</button>
```

55

Browser Object Model (BOM) Window History

Method	Description
back()	Nạp URL trước đó trong danh sách các trang đã duyệt qua.
forward()	Nạp URL đứng sau URL hiện tại trong danh sách các trang đã duyệt qua
go()	Nạp một URL xác định trong danh sách các trang đã duyệt qua.

56

Browser Object Model (BOM)

Window Location

Thuộc tính

Property	Description
Hash	Thiết lập hay lấy phần dữ liệu sau dấu # trong thuộc tính href
host	Thiết lập hay lấy tên máy chủ và số cổng của URL.
hostname	Thiết lập hay lấy phân tên của máy chủ của URL.
href	Thiết lập hay lấy toàn bộ URL như một chuỗi.

Phương thức

Method	Description
assign	Nạp một tài liệu HTML mới.
reload	Nạp lại một trang hiện thời.
replace	Thay thế tài liệu hiện thời bằng cách tải một tài liệu khác với URL xác định.

57