

Páginas WEB Accesibles

JavaScript

Luis Fernando Llana Díaz

Departamento de Sistemas Informáticos y Computación
Universidad Complutense de Madrid

18 de noviembre de 2008

JavaScript, JScript, ECMAScript, DOM

- Netscape introdujo JavaScript (1995).
- Micro\$oft introdujo JScript (1996).
- Compromiso entre Netscape y Micro\$oft: estandarización ECMAScript (1997) (JavaScript 1.5)
- DOM por el W3C, indica como se accede a los elementos de un documento HTML o XML. DOM1 (1998), DOM2 (2000), DOM3 (2004).

- JavaScript no está disponible en todas las plataformas.
- Seguridad: scripts maliciosos.
- Se ejecuta en el cliente:
 - No puede contener secretos para el usuario.
 - El usuario siempre lo puede controlar.

Referencias

- ECMA-262: <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- DOM del W3C: <http://www.w3.org/DOM/DOMTR>
- Mozilla developer: <https://developer.mozilla.org/>

En fichero independiente (el recomendado)

```
<script src="javaScript/codigo.js" type="text/javascript"></script>
```

1

Dentro de HTML

```
<script>  
[.....]  
</script>
```

1

2

3

Tipos Básicos I

Las variables no tienen tipo, los valores sí.

Valores

- Números: 42, 0x3F, 3.14159, 1E-10, NaN, +Infinity, -Infinity.
No hay distinción entre reales y enteros.
- Booleanos: true y false
- Cadenas de caracteres: 'Cadena' o "cadena"
- null.
- undefined.

Arrays

+ concatenación.

Cuidado

```
var a=[1,,67,8,];  
a.length=5;  
a[0] //1  
a[1] // undefined  
a[10]='a' // legal
```

1
2
3
4
5

Cadenas de caracteres

```
var s="hola";  
s[1] // "o"  
s.length // 4
```

1
2
3

Variables

```
var s="3", x;  
  
if (x==undefined) {  
  alert("x no tiene valor");  
} else {  
  alert("x="+x);  
}
```

1
2
3
4
5
6
7

Operadores I

Aritméticos

+, *, /, -, %, ++, --

Bits

&, |, ^, ~, >>, <<

Lógicos

&&, ||, !

Comparación

==, ===, !=, !==, >, <, <=, >=

Operadores II

Cadenas de caracteres

+ concatenación.

Cuidado

```
"10"+3; // "103"  
"10"-3; // 7
```

1
2

```
var s="hola";  
s[1] // "o"  
s.length // 4
```

1
2
3

Instrucciones I

if

```
if (x%2==0) {  
    even();  
} else {  
    odd();  
}
```

1
2
3
4
5

Instrucciones II

switch

```
switch (fruittype) {  
    case "Oranges":  
        s="Oranges are $0.59 a pound.<br>";  
        break;  
    case "Apples":  
        s="Apples are $0.32 a pound.<br>";  
        break;  
    case "Bananas":  
        s="Bananas are $0.48 a pound.<br>";  
        break;  
    case "Cherries":  
        s="Cherries are $3.00 a pound.<br>";  
        break;  
    case "Mangoes":  
    case "Papayas":  
        s="Mangoes and papayas are $2.79 a pound.<br>";  
        break;  
    default:  
        s="Sorry, we are out of " + fruittype + ".<br>";  
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20

Instrucciones III

Bucles

```
var n=5, i=0, k=0, l=1;
while (i<n) {
  k += 1;
  l += 2;
  i++;
}
```

1
2
3
4
5
6

```
var n=200, i=0, k=0, l=1;
do {
  k += 1;
  l += 2;
  i++;
} while (i<n);
```

1
2
3
4
5
6

```
var n=200, k=0, l=1;
for (var i=0; i<n; i++) {
  k += 1;
  l += 2;
}
```

1
2
3
4
5

Instrucciones IV

Acceso a los elementos de un objeto

```
var a = [1,7,,10,25];
a[10]='a';
s="";
for (var i in a) {
  s += ":"+i+":"+a[i]+\n";
}
alert("for in:"+a.length+"\n"+s);
```

1
2
3
4
5
6
7

```
function divs() {
  var d = document.getElementsByTagName("div");
  var s="";
  for (var i in d) {
    s += i+":"+d[i]+\n";
  }
  return s;
}
```

1
2
3
4
5
6
7
8

```
.....
<body onload="alert(divs())">
.....
```

1
2
3

Definición

```
function square(number) {  
  return number * number;  
}
```

1
2
3

- Los parámetros se pasan por valor.
 - Valores primitivos.
 - Objetos (incluyendo arrays);
- Las funciones son variables.

```
var square=function(number) {return number * number;} //alternativa
```

1

```
function map(f,a) {  
  var result=new Array();  
  for (var i = 0; i != a.length; i++)  
    result[i] = f(a[i]);  
  return result;  
}  
a=map(square,a);
```

1
2
3
4
5
6
7

Funciones II

- Se permite recursión.

```
function factorial(n) {  
  if ((n == 0) || (n == 1))  
    return 1;  
  else {  
    var result = (n * factorial(n-1) );  
    return result;  
  }  
}
```

1
2
3
4
5
6
7
8

- El número de argumentos puede ser variable, en la llamada y en la definición

```
function myConcat(separator) {  
  var result = ""; // initialize list  
  // iterate through arguments  
  for (var i = 1; i < arguments.length; i++) {  
    result += arguments[i] + separator;  
  }  
  return result;  
}  
  
// returns "sage. basil. oregano. pepper. parsley."  
myConcat(".", "sage", "basil", "oregano", "pepper", "parsley");
```

1
2
3
4
5
6
7
8
9
10
11

```
function next(n) {  
  if (n==undefined) {  
    return 0;  
  } else {  
    return n+1  
  }  
}  
next() // 0  
next(2) // 3
```

1
2
3
4
5
6
7
8
9

Funciones predefinidas

- `eval`
- `isFinite`
- `isNaN`
- `parseInt` y `parseFloat`
- `Number` y `String`

- `Array`
- `Boolean`
- `Date`
- `Function`
- `Math`
- `Number`
- `RegExp`
- `String`

DOM

- Indica como acceder a cada elemento de HTML, y XML en general (XHTML incluido).
- Estandarizado por el W3C. <http://www.w3.org/DOM/DOMTR>
- Clases importantes:
 - `window`, es parte del llamado DOM0, especificado por Netscape en 1995.
 - `Nodos`: representan elementos y atributos
 - `Documento`: es el nodo que contiene toda la información.
 - `Elementos`: es una subclase de nodo que representan los elementos (de HTML, XML, etc).
 - Cada elemento de HTML tiene una subclase de elemento que representa ese elemento concreto.

<http://antares.sip.ucm.es/~luis/accesibilidadWEB08-09/ejemplos/javascript/dom1.html>

Validación de formularios I

Mal ejemplo

```
<form action="index.php">
<p><label for="login">Login:</label>
<input type="text" name="login" id="login" /></p>
<p><label for="pw">Password:</label>
<input type="password" name="pw" id="pw" /></p>
<p><input type="button" onclick="checkform()" value="send" /></p>
</form>
```

1
2
3
4
5
6
7

```
function checkform() {
var f=document.forms[0];
var error='';
error+= f.login.value==''? 'nlogin' : '';
error+= f.pw.value==''? 'npassword' : '';
if (error!='') {
alert('Por favor ingrese lo siguiente:'+error);
} else {
f.submit();
}
}
```

1
2
3
4
5
6
7
8
9
10
11

Validación de formularios II

Corregido

```
<form action="process.php" method="get" onsubmit="return checkform(this)">
<p><label for="login">Login:</label>
<input type="text" name="login" id="login" /></p>
<p><label for="pw">Password:</label>
<input type="password" name="pw" id="pw" /></p>
<p><input type="submit" value="send" /></p>
</form>
```

1
2
3
4
5
6
7

```
function checkform(f) {
var error='';
error+=f.login.value==''? 'nlogin': '';
error+=f.pw.value==''? 'npassword': '';
if (error!='') {
alert('Please enter the following:'+error);
}
return error=='';
}
```

1
2
3
4
5
6
7
8
9

```
http://antares.sip.ucm.es/~luis/accesibilidadWEB08-09/ejemplos/
javascript/process.php?login=' '&pw=' '
```

Abrir ventanas nuevas

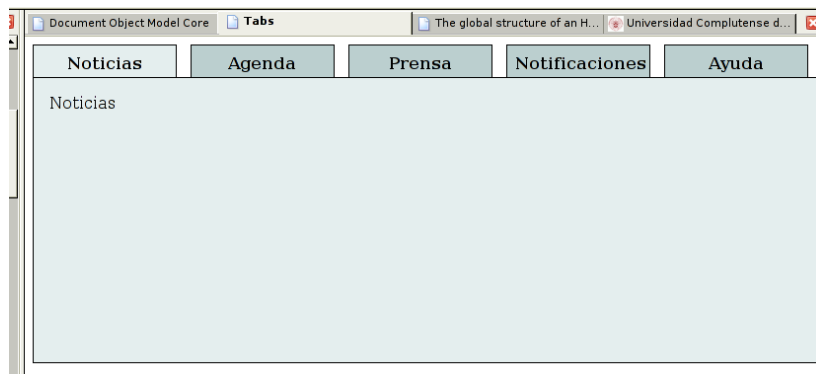
Reglas de accesibilidad

- No abrir ventanas nuevas si avisar previamente.
- Siempre dejar la opción al usuario: nueva ventana, misma ventana,....
- Siempre debe funcionar sin JavaScript.

```
<p><a href="http://www.ucm.es" onclick="link_popup(this); return false">UCM</a></p>
```

```
var strWindowFeatures = 1
"menubar=yes,location=yes,resizable=yes,scrollbars=yes,status=yes,width=400,height=200"; 3
function link_popup(src,features) { 4
  if (features==undefined) { 5
    features=strWindowFeatures; 6
  } 7
  var theWindow = 8
  window.open(src.getAttribute('href'),' ', features); 9
  theWindow.focus(); 10
  return theWindow; 11
} 12
```

Ventanas con pestañas I



- JavaScript puro – tabs.1.php
Se carga todo el código, programación JavaScript.
- Programación en servidor puro (PHP,...) – tabs.2.php
Se carga sólo lo necesario, cada *click* es llamada servidor.
- Mezclando los dos. – tabs.3.php

Menús con JavaScript I

```
<div id="tab" class="tabbing"> 1
  <div id="primero" class="tab activo" > 2
<h1 onClick="javascript:ponActivo(this)"> 3
  Noticias</a></h1> 4
<div class="wrapper"> 5
  <p>Noticias</p> 6
</div> 7
  </div> 8
  <div id="segundo" class="tab noActivo" > 9
<h1 onClick="javascript:ponActivo(this)">Agenda</a></h1> 10
<div class="wrapper"> 11
  <p>Agenda</p> 12
</div> 13
  </div> 14
  ..... 15
  ..... 16
</div> 17
  </div> 18
```

Menús con JavaScript II

```
<style type="text/css">
.....
div.activo div.wrapper{
  visibility : visible;
}
div.noActivo div.wrapper{
  visibility : hidden;
}
.....
</style>
```

1
2
3
4
5
6
7
8
9
10

Menús con JavaScript III

```
function desactivaTodos() {
  tab = document.getElementById("tab");
  divs = tab.getElementsByTagName("div");
  for (i=0; i<divs.length; i++) {
    clase = divs[i].className.toLowerCase();
    if (clase.search("(^| )activo")>0) {
      clase=clase.replace("activo","noActivo");
      divs[i].className=clase;
    }
  }
}
function ponActivo(elemento) {
  desactivaTodos();
  div=elemento.parentNode;
  div.className="tab activo";
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16

Ventanas con pestañas, sólo HTML (con PHP) + CSS I

```
<?php
define(tab,"tab");
define(primer,"primero");
define(segundo,"segundo");
define(tercero,"tercero");
define(cuarto,"cuarto");
define(quinto,"quinto");

if ( !isset($_GET[tab]) ) {
    $tab="primero";
} else {
    $tab=$_GET[tab];
}
}
```

1
2
3
4
5
6
7
8
9
10
11
12
13

Ventanas con pestañas, sólo HTML (con PHP) + CSS II

```
function clase($id) {
    global $tab;
    $clase="class=\"tab ";
    if ( $tab==$id ) {
        $clase.="activo";
    } else {
        $clase.="noActivo";
    }
    $clase.="\"";
    $clase.=" id=\"".$id."\" ";
    return $clase;
}
?>
```

1
2
3
4
5
6
7
8
9
10
11
12
13

Ventanas con pestañas, sólo HTML (con PHP) + CSS III

```
<div id="tab" class="tabbing">
  <div <?=clase(primeros)?> >
<h1><a href="tabs.2.php?tab=primero" >Noticias</a></h1>
<?php if ($tab==primero) { ?>
<div class="wrapper">
  <p>Noticias</p>
</div>
<?php }?>
</div>
  <div <?=clase("segundo")?> >
<h1><a href="tabs.2.php?tab=segundo" >Agenda</a></h1>
<?php if ($tab==segundo) {?>
<div class="wrapper">
  <p>Agenda</p>
</div>
<?php }?>
  .....
  .....
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

Ventanas con pestañas, con todo I

```
<div <?=clase("primero")?> >
<h1><a href="tabs.3.php?tab=primero"
  onClick="javascript:ponActivo(this); return false">
  Noticias</a></h1>
<div class="wrapper">
  <p>Noticias</p>
</div>
</div>

  <div <?=clase("segundo")?> >
<h1><a href="tabs.3.php?tab=segundo"
  onClick="javascript:ponActivo(this); return false">Agenda</a></h1>
<div class="wrapper">
  <p>Agenda</p>
</div>
</div>
  .....
  .....
```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18