

Sistemas Informáticos

Tema 2: Sistemas distribuidos basados en WWW

2.4 Ejecución de código en el cliente: javascript, ajax, jquery



Javascript - Introducción

- En su uso típico, enriquece las páginas HTML incorporando características dinámicas e interactivas
- Se ejecuta en cliente (en el navegador web)
- Usos comunes (algunos ejemplos)
 - Inclusión de adornos y efectos visuales en textos e imágenes de las páginas web
 - Manipulación de contenidos o aspecto de forma dinámica
 - Realización de operaciones matemáticas sencillas
 - Validación de datos introducidos en formularios
 - Gestión del sistema de navegación (menús desplegables)
 - Control del tipo y versión del navegador web, uso de la fecha y hora actuales, verificación de plugins...



Javascript - Introducción

- Características
 - Diseñado para manejar interacciones en el navegador web
 - Con sintaxis muy similar a Java (y a C)
 - Orientado a objetos: objetos, propiedades, métodos, eventos
 - Código interpretado
 - En el caso de ejecutarse en el cliente, por el motor Javascript del navegador
- Guía de referencia
 - <http://www.w3schools.com/js/default.asp>



Javascript - Introducción

- Lenguaje para la creación de rutinas en páginas HTML
- Desarrollado por Netscape en colaboración con Sun, aunque en la actualidad cuando hablamos de JavaScript nos referimos a ECMAScript, una especificación basada en la definición de Netscape
- Proporciona objetos para manejar un documento HTML:
 - Window* •*Location* •*Button*
 - Document* •*Navigator* •*Textarea*
 - Frame* •*History* •*Radio*
- Permite capturar eventos:
 - click* •*focus* •*mouseOver*
 - change* •*load* •*select*

JavaScript - Introducción

- **DHTML** = Dynamic HTML
 - HTML + JavaScript + CSS
 - NO tiene nada que ver con el proceso de generación dinámica de páginas web en el servidor (mediante Perl, PHP, JSP, ASP, etc.)
- **AJAX** = Asynchronous JavaScript And XML
 - JavaScript + comunicación cliente-servidor asíncrona
 - Originalmente XML, actualmente más JSON
 - Aplicaciones interactivas (gracias a JavaScript) y asíncronas
 - Google Web Toolkit (GWT): *framework* creado por Google que permite ocultar la complejidad de aspectos de la tecnología AJAX
 - jQuery es otra biblioteca



Ejemplo Javascript

```
<HTML>
<HEAD>
<TITLE>Ejemplos Javascript: ejemplo práctico </TITLE>

<script language="JavaScript">
<!--
var id,pause=0,position=0;
function scorrevole() {
var i,k,msg=" Tres tristes tigres ... - ";
k=(100/msg.length)+1;
for(i=0;i<=k;i++) msg+=" "+msg;
document.form2.scorrevole.value=msg.substr(position,position+100);
if(position++==100) position=0;
id=setTimeout("scorrevole()",100); } </HEAD>
//-->
</script>                                <BODY bgcolor="white" onload="scorrevole()">

                                <form name=form2><input type="text"
                                name="scorrevole" size="40"></form>

                                </body>
                                </html>
```



Características

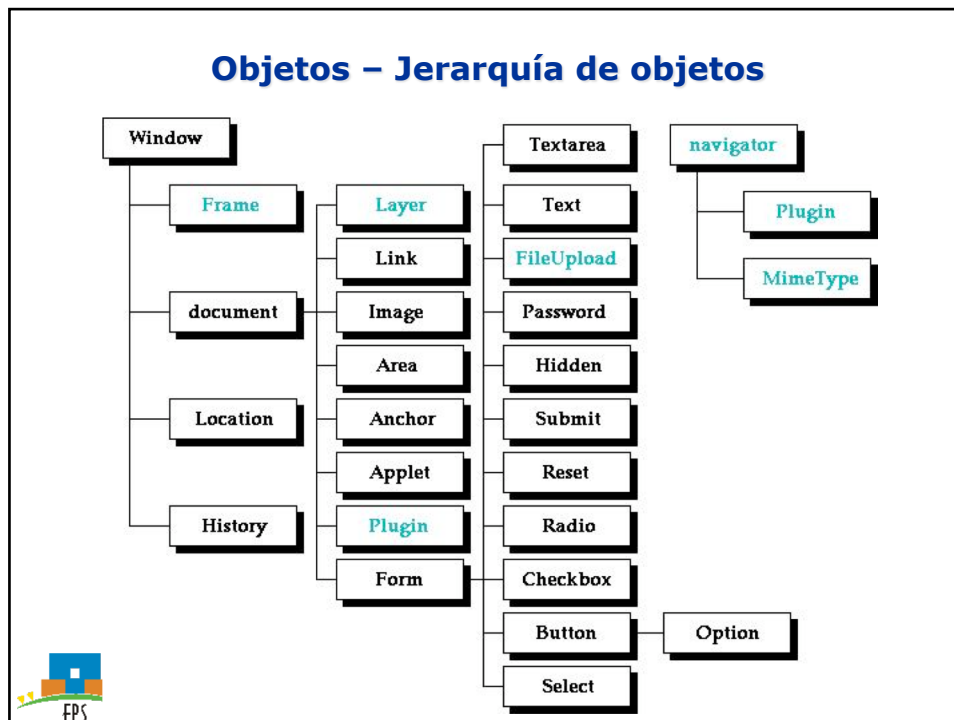
- Enfocado al manejo de elementos de documentos HTML y el navegador web.
- Objetivo: disminuir el trasiego de datos entre el cliente y el servidor
 - Capturar los eventos generados por el usuario, y responder a ellos sin depender del servidor
 - Realizar cálculos sencillos sin necesidad de comunicación con el servidor
 - Comprobar los datos que el usuario introduce en un formulario antes de enviarlos al servidor
- Interpretado
 - No se compila
 - No se ejecuta de forma independiente, como aplicación autónoma
 - Se inserta en el código HTML, y es interpretado por el navegador web
- Orientado a objetos
 - Peculiaridad: los objetos no tienen clases definidas
- Sensible a mayúsculas y minúsculas
- Fin de sentencia con ; o con salto de línea
 - `x = 2`
 - `x = 2;`



Variables

- `var x = valor`
- No tiene tipado fuerte (en sentido de lenguajes como C o Java), sino tipado dinámico o débil:
 - `x = 2` // El tipo de datos de x es entero, pero no se declara
 - `x = "hola"` // El tipo de x se puede cambiar
- Siete tipos de datos:
 - Number
 - String
 - Boolean
 - Null
 - Undefined
 - Symbol
 - Object
- Ejemplos
 - `x = 1.7` // Número
 - `x = "hola"` // Cadena de caracteres
 - `x = 'hola'` // Cadena de caracteres
 - `x = 'hola' + 'mundo'` // Concatenación de cadenas de caracteres
 - `x = true` // Booleano
 - `x = null` // Literal nulo
 - `x = [2, 3, 5, "Alice"]` // Array, `x[0]`, `x[1]`, `x[2]`, `x[3]`
 - `x = { a:2, b:"Bob" }` // Array asociativo, `x['a']`, `x['b']`





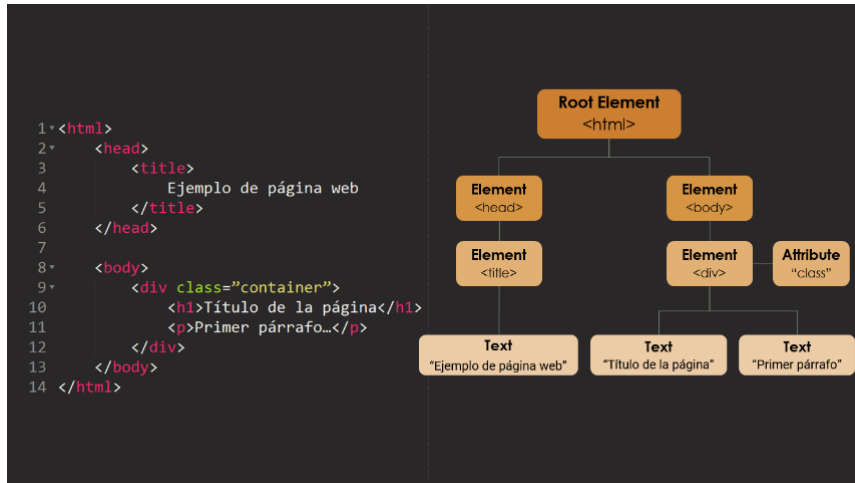
Objetos II

- **Objeto = propiedades + métodos (+ eventos)**
- Propiedades
 - Características (atributos, elementos) de un objeto
 - Pueden ser a su vez objetos
 - Creación: `profesor = new Object(); profesor.nombre = "Kowalski";`
 - Ejemplos: `unaCadena.length`, `window.document`
- Métodos
 - Funcionalidades (funciones) de un objeto
 - Ejemplos: `unaCadena.substring(2)`, `window.write('<p>Hola mundo</p>')`
- Eventos
 - Acciones que ocurren en la interfaz gráfica sobre un objeto de la misma
 - Debidas al usuario: click de ratón, pulsación de tecla, etc.
 - Asociadas al documento: carga del contenido, minimización/maximización de la ventana, etc.
 - Su ejecución se utiliza para la invocación de métodos
 - Ejemplos:


```
<input type="button" onClick="window.alert('¡Botón pulsado!')">
```



Navegación por el árbol DOM (i)

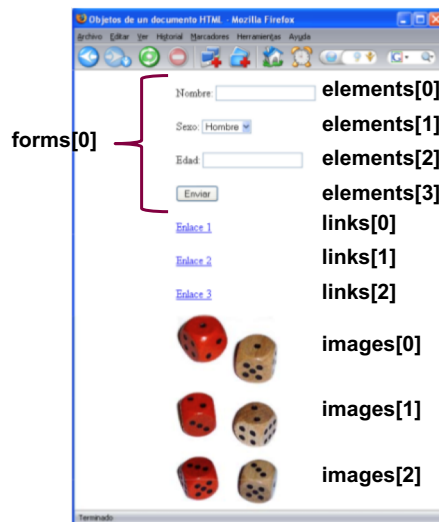


Navegación por el árbol DOM (ii)

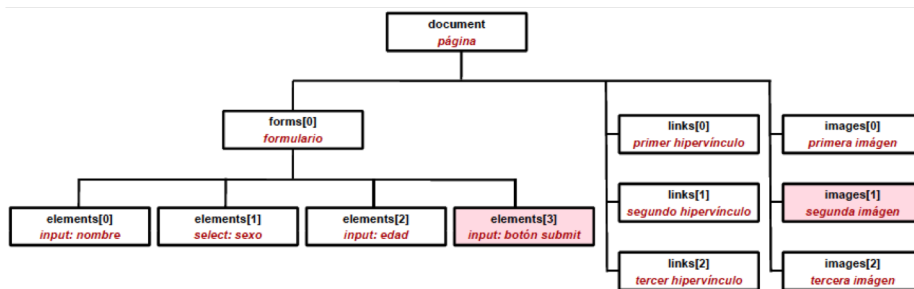
- `document.getElementById("id")`
- `document.getElementsByTagName("tag")`
- `document.getElementsByClassName("clase")`
- También proporciona métodos para:
 - Crear, eliminar, sustituir, concatenar... nodos del árbol
 - Recorrer el árbol en sus distintas dimensiones:
 - Elementos: hijos, hermanos, ancestros...
 - Atributos
 - Textos



Identificación de objetos (i)



Identificación de objetos en HTML



Cuarto elemento del formulario (botón Submit):

`document.forms[0].elements[3]`

`document.getElementById("botón submit")`

`document.getElementsByTagName("input")[2]`

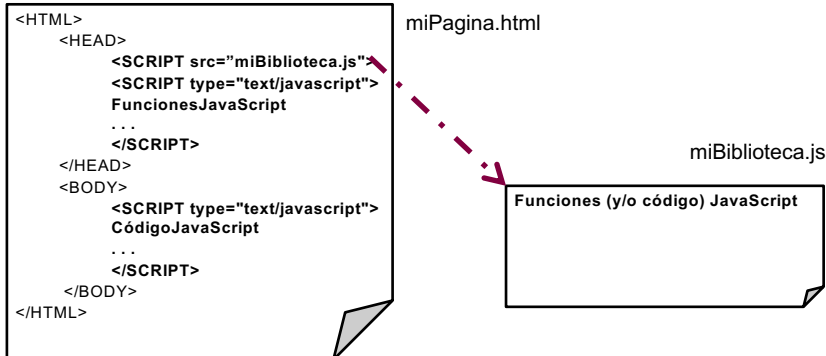


Se supone que está así definido el botón de submit:

`<INPUT id="botón submit" type="submit" >`

Inserción de código (i)

- Las funciones se suelen escribir en la cabecera <head> del documento HTML, o en fichero de texto independiente (normalmente con extensión js).
- Código directo se puede escribir en el cuerpo del documento



Inserción de código (ii)

- Ejemplo: código directo

```

<HTML>
  <HEAD>
  </HEAD>
  <BODY>
    <SCRIPT type="text/javascript">
      document.write("<H1><U>Títulos</U><BR></H1>")
      for (i=1; i<=6; i++) {
        document.write("<H" + i + ">Título " + i)
        document.write("</H" + i + "><BR>")
      }
    </SCRIPT>
  </BODY>
</HTML>
  
```



Inserción de código (iii)

- Ejemplo: función en la cabecera

```
<HTML>
<HEAD>
  <SCRIPT type="text/javascript">
    function mostrarTitulos() {
      document.write("<H1><U>Títulos</U><BR></H1>")
      for (i=1; i<=6; i++) {
        document.write("<H" + i + ">Título " + i)
        document.write("</H" + i + "><BR>")
      }
    }
  </SCRIPT>
</HEAD>
<BODY>
  <INPUT type="button" value="Mostrar" onClick="mostrarTitulos()">
</BODY>
</HTML>
```



Inserción de código (iv)

- Ejemplo: función en un fichero independiente

```
<HTML>
<HEAD>
  <SCRIPT src="mostrar-titulos.js"></SCRIPT>
</HEAD>
<BODY>
  <INPUT type="button" value="Mostrar" onClick="mostrarTitulos()">
</BODY>
</HTML>
```

```
function mostrarTitulos() {
  document.write("<H1><U>Títulos</U><BR></H1>")
  for (i=1; i<=6; i++){
    document.write("<H" + i + ">Título " + i)
    document.write("</H" + i + "><BR>")
  }
}
```



Ejemplo práctico 1

- Entrada salida de datos mediante pop-ups

```
<HTML>
  <BODY>
    <SCRIPT type="text/javascript">
      var num1, num2, sum
      num1 = window.prompt("Introduce el primer numero")
      num2 = window.prompt("Introduce el segundo numero")
      sum = window.parseInt(num1) + window.parseInt(num2)
      window.alert("Suma = " + sum)
    </SCRIPT>
  </BODY>
</HTML>
```



Ejemplo práctico 2

- Evento de click

```
<HTML>
  <HEAD>
    <SCRIPT type="text/javascript">
      function identificarBoton(evento) {
        if(evento.button==0 || evento.button==1){
          window.alert("Has hecho clic con el boton izquierdo") }
        else {
          window.alert("Has hecho clic con el boton derecho")
        }
      }
    </SCRIPT>
  <HEAD>
  <BODY onmouseup="identificarBoton(event)" onclick="identificarBoton(event)">
    <H1>Haz click donde quieras!</H1>
  </BODY>
</HTML>
```



Ejemplo práctico 3

- Evento de carga del documento.
- Eventos de acceso a los objetos con el ratón

```
<HTML>
  <HEAD>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <TITLE>Intercambio de imágenes</TITLE>
  </HEAD>
  <BODY onload="window.alert('¡Bienvenido! La página se ha cargado!')">
    <IMG id="unalmagen" src="bob1.jpg"
      onmouseover="document.unalmagen.src='bob2.jpg'"
      onmouseout="document.unalmagen.src='bob1.jpg'"/>
  </BODY>
</HTML>
```



Ejemplo práctico 4

- Evento del foco

```
<HTML>
<HEAD>
<TITLE> Color de Fondo </TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<CENTER>
<H1>Cambio del color de fondo</H1><BR>
<FORM>
<INPUT type="text" name="bgr" value="rojo" onfocus="document.bgColor='red'"><BR>
<INPUT type="text" name="bgg" value="verde" onfocus="document.bgColor='green'"><BR>
<INPUT type="text" name="bgb" value="azul" onfocus="document.bgColor='blue'"><BR>
<INPUT type="text" name="bgy" value="amarillo" onfocus="document.bgColor='yellow'"><BR>
</FORM>
</CENTER>
</BODY>
</HTML>
```



Ejemplo práctico 5

- Acceso a un objeto por ID

```
<HTML>
<HEAD>
  <SCRIPT type="text/javascript">
    function factorial(n) {
      if (n < 2) { return 1}
      else { return n * factorial(n-1)}
    }
  </SCRIPT>
</HEAD>
<BODY>
  <FORM>
    Entrada:&nbsp;
    <INPUT id="argumento" type="text"
      onchange="res=factorial(this.value);
      document.getElementById('resultado').value = res;"/>
    <BR><BR> Salida:&nbsp;
    <INPUT id="resultado" type="text" />
  </FORM>
</BODY>
</HTML>
```



Ejemplo práctico 6

- Acceso a un objeto por etiqueta (tag)

```
<HTML>
<HEAD>
  <STYLE>
    .clase1{text-decoration:none;color:red;}
    .clase2{text-decoration:none;color:green;}
    .clase3{text-decoration:none;color:blue;}
  </STYLE>
  <SCRIPT type="text/javascript">
    function cambiarEstilo(estilo){
      var lista=document.getElementById("miLista");
      var items=lista.getElementsByTagName("li");
      for(i=0; i<items.length; i++){
        items[i].getElementsByTagName("a")[0].setAttribute("class",estilo);
      }
    }
  </SCRIPT>
</HEAD>
```



Ejemplo práctico 6 (ii)

- Acceso a un objeto por etiqueta (tag)

```
<BODY>
  <H1><UL id="miLista">
    <LI><A href="javascript:cambiarEstilo('clase1');">Estilo1</A></LI>
    <LI><A href="javascript:cambiarEstilo('clase2');">Estilo2</A></LI>
    <LI><A href="javascript:cambiarEstilo('clase3');">Estilo3</A></LI>
  </UL></H1>
</BODY>
</HTML>
```



Ejemplo práctico 7

- Creación y comunicación entre ventanas

```
<HTML>
<HEAD>
<SCRIPT type="text/javascript">
  function crearVentana(){
    // Leemos el contenido HTML de la ventana
    var p=window.document.getElementById("ventanaEntrada");
    // Creamos la ventana de entrada
    var v=window.open("", "", "width=300,height=200,menubar=no,
      location=no,status=no,resizable=no");
    var d=v.document;
    d.open();
    d.write(p.innerHTML);
    d.close();
  }
  function salir(texto){
    window.document.getElementById("textoSalida").value = texto;
  }
</SCRIPT>
</HEAD>
```



Ejemplo práctico 7

- Creación y comunicación entre ventanas

```
<BODY>
<DIV id="ventanaEntrada" style="display:none;">
Introduce texto de entrada:
<BR>
<TEXTAREA id="textoEntrada" rows="8" cols="32"></TEXTAREA>
<BR>
<A href="javascript:opener.salir(window.document.getElementById
('textoEntrada').value); close();">Guardar</A>
<A href="javascript:opener.salir(""); close();">Borrar</A>
</DIV>
<DIV>
<A href="javascript:crearVentana();">Introducir texto...</A> <HR>
</DIV>
<DIV>
Texto introducido: <BR>
<TEXTAREA id="textoSalida" rows="8" cols="32" readonly></TEXTAREA>
<HR>
</DIV>
</BODY>
</HTML>
```



Ocultar el código

- Si el navegador no reconoce javascript, el código será mostrado como parte del contenido de la página.
- Para evitarlo:

```
<html>
<body>
<script type="text/javascript">
<!--
document.getElementById("demo").innerHTML=Date();
//-->
</script>
</body>
</html>
```



Ejercicio de validación de formulario

Realizar un document HTML con el siguiente formulario:

Nombre:

Edad:

E-mail:

No se enviarán datos al servidor hasta que:

- El campo para el nombre tenga un “nombre” válido de al menos 2 caracteres, no permitiéndose caracteres en blanco ni al principio ni al final
- El campo edad tenga una edad válida
- El campo e-mail tenga una dirección válida (al menos contenga character @)

El formulario se enviará para su procesamiento a uno de los siguientes servicios <https://postman-echo.com/get> o <https://postman-echo.com/post>, el primero acepta peticiones en modo GET y el segundo en modo POST. Comparar el resultado obtenido en ambos casos te ayudará a comprender la diferencia entre ambos.

