



TEMA 4

PROGRAMACION DE LADO DEL CLIENTE

Ing. Carlos David Montellano Barriga



CONTENIDO



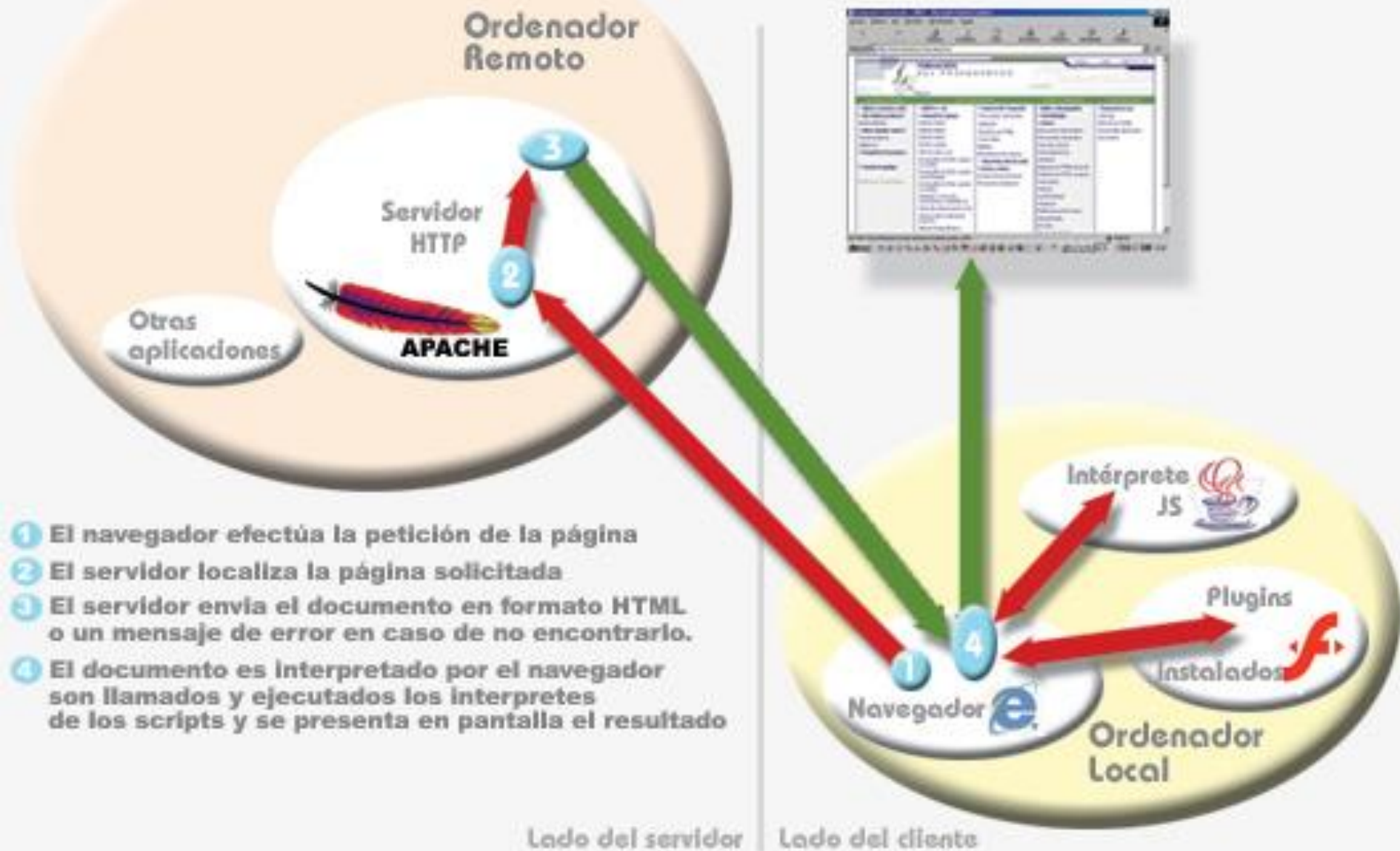
- Programación del lado del cliente.
- El Lenguaje JavaScript
 - JavaScript básico
 - Java Script Avanzado
- DOM (Document Object Model)
- Eventos
- Validación de entrada de datos del lado del cliente.
- Soporte del navegador.
- Frameworks y Librerías
- Framework JQuery
- JQuery Mobile

PROGRAMACIÓN DEL LADO DEL CLIENTE



- Es la programación de scripts y programas, que se ejecutan en el **cliente web o navegador**, para crear aplicaciones enriquecidas del lado del cliente.
- Un lenguaje del lado cliente es totalmente independiente del servidor
- Permite que la página pueda ser albergada en cualquier sitio.
- Muchas lenguajes de programación del cliente requieren tener instalados los ***plug-in*** adecuados para mostrarlos correctamente .

Páginas dinámicas usando únicamente aplicaciones del lado del cliente



ALGUNOS LENGUAJES DE PROGRAMACIÓN DEL LADO DEL CLIENTE



- Javascript
- VBasicScript
- Flash
- Applets de Java
- Silver Ligth



EL LENGUAJE JAVASCRIPT

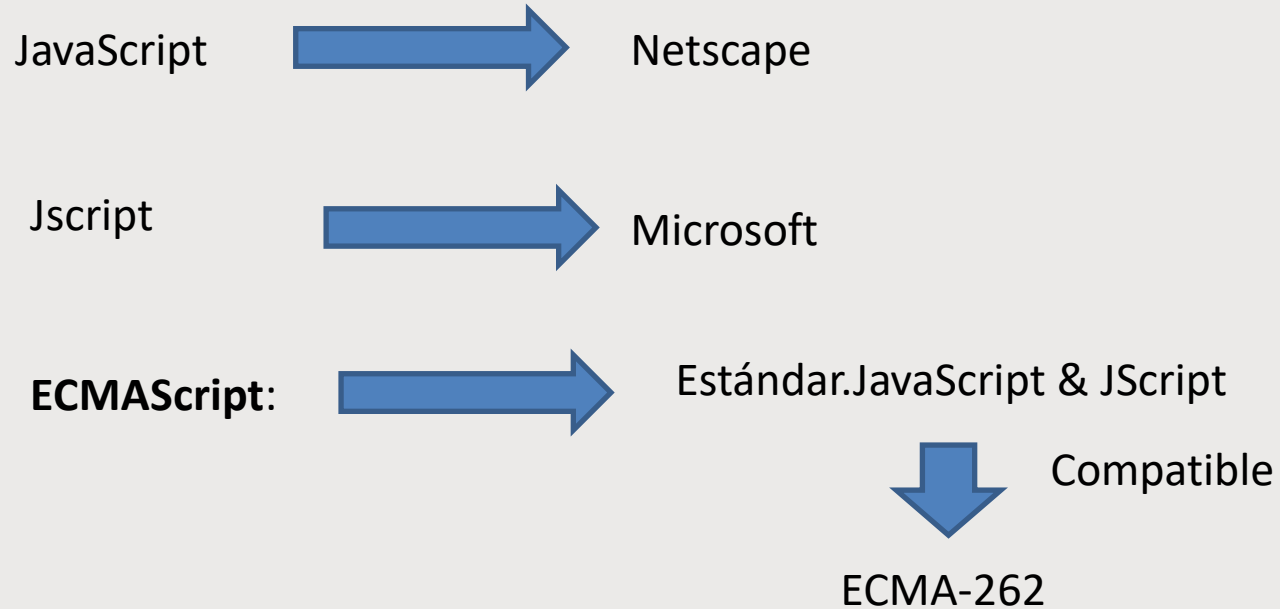
- ✓ Creado por Brendan Eich para Netscapa aparece en Netscape 2.0B3 en 1995.
- ✓ Es un lenguaje de programación interpretado con base en la sintaxis del lenguaje C.
- ✓ Está basado en objetos y guiado por eventos.
- ✓ No tiene nada que ver con Java
- ✓ No requiere de plug-ins para funcionar



Brendan Eich
Creador del Javascript



ORIGEN JAVASCRIPT



CAPACIDADES DE JAVASCRIPT



- Abrir nuevas ventanas controlando su tamaño, look & feel, controles, etc.
- Incorporar validaciones a los formularios.
- Efectos visuales en la página.
- Generar respuestas ante distintos eventos
- Crear, Eliminar o cambiar atributos de elementos de una página HTML en forma dinámica.
- Crear o Leer Cookies
- Detectar la configuración del Browser.

HOLA MUNDO EN JAVA SCRIPT



HolaMundo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Primer Javascript</title>
<script type="text/javascript" language="javascript">
alert("Hola Mundo");
</script>
</head>
<body>
</body>
</html>
```

COMO INCLUIR EL JAVASCRIPT



1. Dentro del elemento Script. Por ejemplo:

```
<script language="JavaScript" type="text/JavaScript">  
  
</script>
```

2. En archivos independientes.

```
<script language="JavaScript" src="archivo.js">  
</script>
```

3. Se puede incluir directamente en un evento asociado a algún elemento del documento.

```
<input type="button" onclick="alert('Gracias por su click');return false;" value="Click">
```

DONDE COLOCAR LOS SCRIPT



- Los scripts se colocan en...
 - Dentro de `<head> ... </head>`
 - Si genera una salida, dentro de `<body> ... </body>`
 - Si hace referencia a un elemento HTML, después de dicho elemento

VARIABLES



- Son case sensitive
 - Se recomienda empezar por letra o guión
 - No deben de coincidir con palabras reservadas
- Declaración:
 - Asignándole un valor: **x = 42**
 - Con la palabra reservada “var”: **var x**
 - O bien ambos: **var x = 42**
- Las variables pueden ser:
 - Locales o globales
 - Si se introduce una variable en una función y no se declara, !ES GLOBAL!

TIPOS DE DATOS



- Tipo de datos
 - Numérico (entero, real, etc)
 - Booleano (true o false)
 - String (“Hola Mundo”)
 - Null (Valor nulo)
 - NaN (Valor indefinido)
 - Object
- Tipado dinámico
 - No es necesario declarar el tipo de la variable
 - Se pueden convertir de un tipo a otro durante la ejecución (no se suele hacer porque trae problemas)

Operadores aritmeticos	
+	Suma
-	Resta
*	Multiplicacion
/	Division
%	Modulo
++	Ingremento
--	Decremento

Operadores Comparacion	
==	Igual
!=	Diferente
>	Mayor
<	Menor
>=	Mayor o igual
<=	Menor o igual

Operadores de asignacion	
=	Igual
+=	Adionar y asigna
-=	Resta y asigna
*=	Multiplica y asigna
/=	Divide y asigna
%=	Modulo y asigna

Operadores Logicos	
&&	Y
	O
!	no

EJEMPLO

variables.html

```
<script type="text/javascript" language="javascript">
var iva = 16;    // variable tipo entero
document.writeln(iva+"<br>");
var total = 234.65; // variable tipo decimal
document.writeln(total+"<br>");
var mensaje = "Bienvenido a nuestro sitio web";
document.writeln(mensaje+"<br>");
var nombreProducto = 'Producto ABC';
document.writeln(nombreProducto);
document.write("<br>");
var letraSeleccionada = 'c';
document.writeln(letraSeleccionada);
document.write("<br>");
var texto1 = "Una frase con 'comillas simples' dentro";
document.writeln(texto1);
document.write("<br>");
var texto2 = 'Una frase con "comillas dobles" dentro';
document.writeln(texto2);
document.write("<br>");
</script>
```

POP UPS



- Alerta
 - alert("algún texto")
- Confirmación
 - confirm("algún texto")
- Introducir datos
 - prompt("según texto","valor por defecto")

popus.html

```
<script type="text/javascript">
a=prompt("Introduzca el valor de a");
b=prompt("Introduzca el valor de b");
c=parseInt(a)+parseInt(b);
alert("el valor de a+ b es "+c);
if (confirm("Desea confirmar lo hecho")==true)
{ alert("confirmando"); }
else
{ alert("cancelando"); }
</script>
```

CONDICIONALES

```
if (<condicion 1>)  
{ <declaracion 1>; }  
    Else  
{ <declaracion 2>; }
```

mayor.html

```
<html>  
  <body>  
    <script type="text/javascript">  
      a = 8;  
      b = 3;  
      if (a < b)  
      {  
        document.write("a es menor que b");  
      }  
      else  
      {  
        document.write("a no es menor que b");  
      }  
    </script>  
  </body>  
</html>
```

switch (<expresion>)

```
{  
    case <literal o tipo>: <declaraciones >;  
                          [break;]  
    case <literal o tipo>: <declaraciones >;  
                          [break;]  
    default: <declaraciones >;  
}
```

casos.html

```
<html>  
  <body>  
    <script type="text/javascript">  
      posicion = "arriba";  
      switch(posicion) {  
        case "arriba": // Bloque 1  
          document.write("La variable contiene");  
          document.write(" el valor arriba");  
          break;  
        case "abajo": // Bloque 2  
          document.write("La variable contiene");  
          document.write(" el valor abajo");  
          break;  
        default: // Bloque 3  
          document.write("La variable contiene otro  
valor");  
          document.write(" distinto de arriba y abajo");  
      }  
    ?>  
  </body>  
</html>
```

BUCLES



```
while (condicion){  
  //código  
}
```

```
for (inicializacion; condicion; incremento){  
  //código  
}
```

```
do {  
  //código  
}  
while (condicion)
```

FUNCIONES

```
<script type="text/javascript">
var numero = prompt("Introduce un número entero");
var resultado = parImpar(numero);
alert("El número "+numero+" es "+resultado);
function parImpar(numero) {
    if(numero % 2 == 0) {
        return "par";
    }
    else {
        return "impar";
    }
}
</script>
```

par.html

- Sintaxis

function mifuncion(argument1,argument2,etc)

{

//Codigo

}

- Se puede usar la palabra reservada **return** pero no hay que indicar nada en la declaración de la función

COMENTARIOS



- Una línea:

// comentario de una línea

- Varias líneas:

/*

comentario de varias líneas

*/

OBJETOS NATIVOS DE JAVASCRIPT



- **La clase Array**

- var variable1 = new Array();
 - var variable2 = new Array(7);
 - var variable3 = new Array(2, "hola", true, 45.34);
- length, calcula el número de elementos de un array

arreglo.html

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado",  
"Domingo"];  
for(i in dias) {  
  alert(dias[i]);  
}
```

CLASE DATE



saludo.html

```
function saludar(){  
  var tiempo = new Date();  
  var hora, cad="son las ";  
  with (tiempo){  
    hora = getHours();  
    cad += hora + ":" + getMinutes()+":"+getSeconds();  
  }  
  if (hora < 12)  
    cad = "Buenos días, " + cad;  
  else if (hora < 18)  
    cad = "Buenas tardes, " + cad;  
  else  
    cad = "Buenas noches, " + cad;  
  return cad  
}
```

MANEJO DE CADENAS DE CARACTERES



- El objeto String se usa para manipular cadenas de caracteres. En JavaScript todo texto encerrado entre comillas, dobles o simples, se interpreta como una cadena.
 - Propiedades
 - length***: devuelve la longitud de la cadena.
 - prototype***: permite agregar métodos y propiedades al objeto.
 - Metodos
 - `charAt()` `toUpperCase`
 - `substring(,)` `cadena.toLowerCase()`
 - `cadena.indexOf()`



cadenas.html

```
<script type="text/javascript">
  var cadena=prompt('Ingrese una cadena:', '');
  document.write('La cadena ingresada es:'+cadena);
  document.write('<br>');
  document.write('La cantidad de caracteres son:'+cadena.length);
  document.write('<br>');
  document.write('El primer carácter es:'+cadena.charAt(0));
  document.write('<br>');
  document.write('Los primeros 3 caracteres son:'+cadena.substring(0,3));
  document.write('<br>');
  if (cadena.indexOf('hola')!=-1)
    document.write('Se ingresó la subcadena hola');
  else
    document.write('No se ingresó la subcadena hola');
  document.write('<br>');
  document.write('La cadena convertida a mayúsculas es:'+cadena.toUpperCase());
  document.write('<br>');
  document.write('La cadena convertida a minúsculas es:'+cadena.toLowerCase());
  document.write('<br>');
</script>
```

EXPRESIONES REGULARES



- Las expresiones regulares son modelos que describen las combinaciones de caracteres en el texto.
- una serie de caracteres que forman un patrón

checkmail.html

```
<SCRIPT TYPE="text/javascript" LANGUAGE="JavaScript">
<!--
function checkMail(cadena) {
var plant = /^[^\\w^@^\\.^~]+/gi
if (plant.test(cadena))
alert(cadena + " contiene caracteres extraños.")
else{
plant = /(^[\\w+])(@{1})([\\w\\.~]+)$/i
alert(cadena + " es correcta.")
else
alert(cadena + " no es válida.")
}
}
// -->
if (plant.test(cadena))
/SCRIPT
```

EJEMPLOS EXPRESIONES



Cualquier letra en minúscula	[a-z]
Entero	^(?:\+ -)?\d+\$
Correo electrónico	/[\w-\.]{3,}@([\w-]{2,}\.)*([\w-]{2,}\.){2,4}/
URL	^(ht f)tp(s?)\:\V\V[0-9a-zA-Z]([-\.\w]*[0-9a-zA-Z])*(:(0-9)*)*(\V?)([a-zA-Z0-9\-\.\?\\\'\\\/\\\\+&%\\$#_])*?\$
Contraseña segura	(?!^[0-9]*\$)(?!^[a-zA-Z]*\$)^[a-zA-Z0-9]{8,10}\$ (Entre 8 y 10 caracteres, por lo menos un dígito y un alfanumérico, y no puede contener caracteres espaciales)
Fecha	^\d{1,2}\V\d{1,2}\V\d{2,4}\$ (Por ejemplo 01/01/2007)
Hora	^(0[1-9] 1\d 2[0-3]):([0-5]\d):([0-5]\d)\$ (Por ejemplo 10:45:23)
Número tarjeta de crédito	^((67\d{2}) (4\d{3}) (5[1-5]\d{2}) (6011))(-?\s?\d{4}){3}((3[4,7])\d{2}-?\s?\d{6}-?\s?\d{5})\$
Número teléfono	^[0-9]{2,3}-? ?[0-9]{6,7}\$

CREACIÓN DE OBJETOS NUEVOS



- Es posible definir nuevos objetos de dos formas diferentes:
 - Mediante inicializadores de objeto (dando el valor inicial del mismo)
- Mediante una función constructora y “new” (lo más recomendable)

```
function coche(marca, modelo, color) {  
  this.marca = marca  
  this.modelo = modelo  
  this.color = color  
}  
  
{  
  coche1 = new coche("Seat", "Toledo", "rojo");  
}
```

DEFINICIÓN DE MÉTODOS



- Vinculación Externa

crearobjeto.html

```
<script type="text/javascript">
function coche(marca, modelo, color) {
  this.marca = marca
  this.modelo = modelo
  this.color = color
  this.mostrar = MostrarCoche;
}
function MostrarCoche () {
  alert("El autos marca "+this.marca+" y modelo
"+this.modelo);
}
</script>
}
```

- Prototype

crearobjetoprototype.html

```
<script type="text/javascript">
function coche(marca, modelo, color) {
  this.marca = marca
  this.modelo = modelo
  this.color = color
}
Coche.prototype. MostrarCoche=function () {
  alert("El autos marca "+this.marca+" y modelo
"+this.modelo);
}
</script>
}
```

JSON



- Consiste en asignarle a una variable un literal de objeto:
 - `nomObj = { prop1: val1, prop2: val2, ...,propN: valN }`
- Donde
 - `nomObj` es el nombre del nuevo objeto
 - `prop1,...,propN` son identificadores que representan las propiedades del objeto
 - `val1,...,valN` son los valores asignados a cada propiedad del objeto

```
coche={marca:"Ford", modelo:"Focus", color:"Rojo"}  
document.write("El coche es: "+ coche.marca)
```



- ```
{
 "employees": [
 {"firstName": "John", "lastName": "Doe"
 },
 {"firstName": "Anna", "lastName": "Smith"
 },
 {"firstName": "Peter", "lastName": "Jones"
 }
]
}
```



- JSON Objects
- JSON objects are written inside curly braces.
- Just like in JavaScript, objects can contain multiple name/value pairs:
- `{"firstName": "John", "lastName": "Doe"}`



# DOM DOCUMENT OBJECT MODEL





# QUE ES DOM

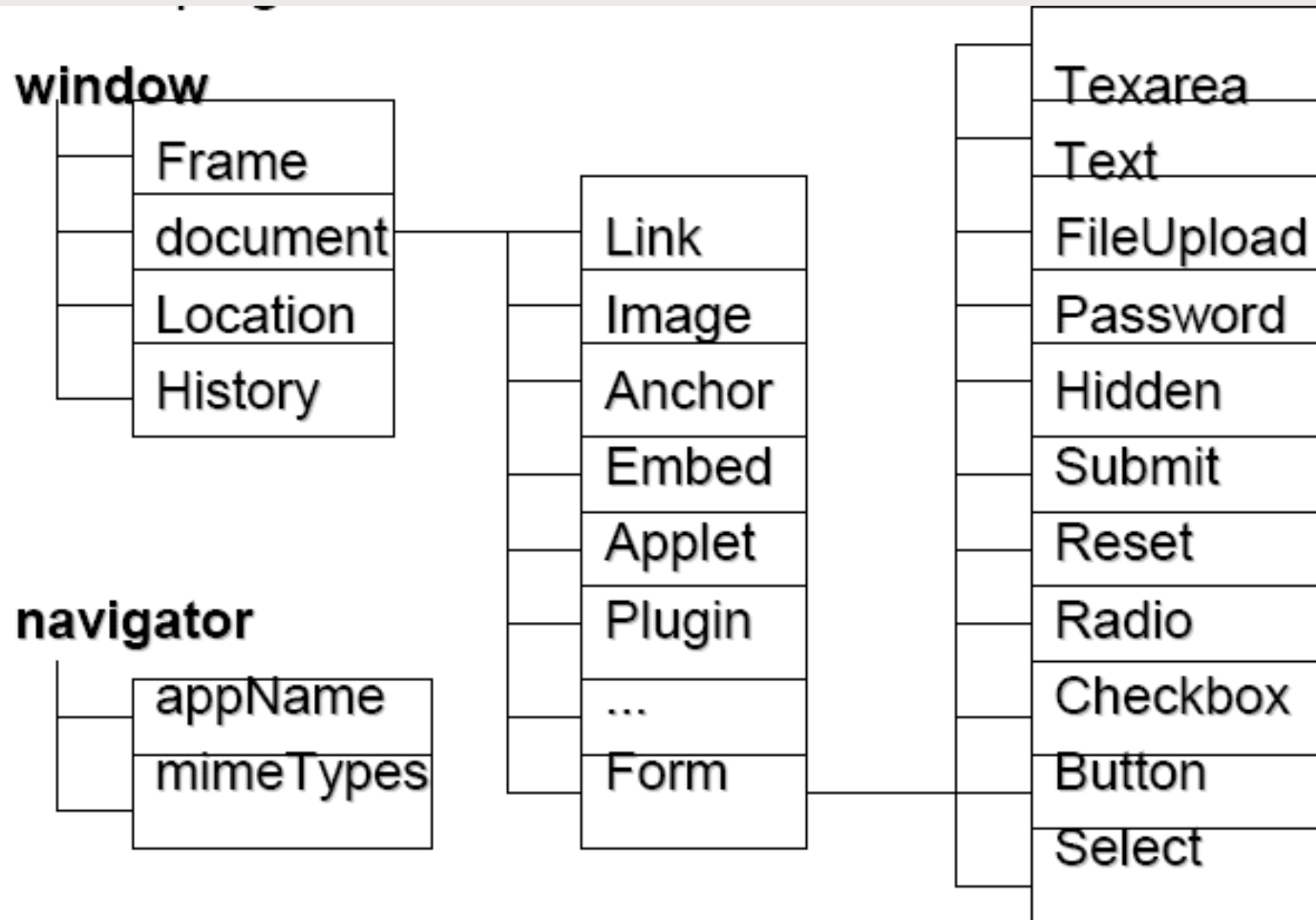


- Dom es una plataforma o interfaz neutral que permite a los programas y scripts acceder o modificar el contenido, estructura y estilo de los documentos.
- El DOM permite un acceso a la estructura de una página HTML mediante el mapeo de los elementos de esta página en un árbol de nodos

# DOM



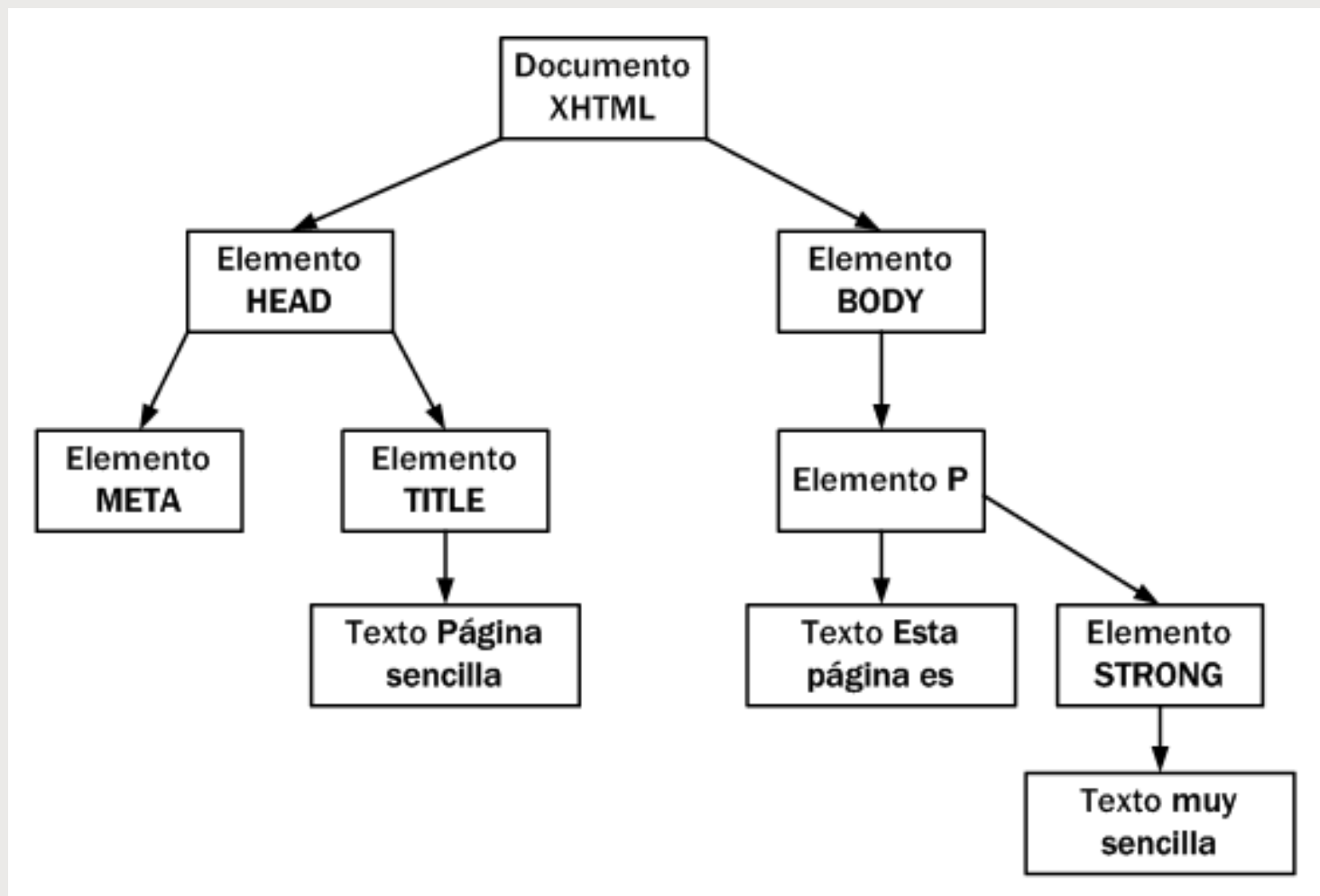
Cuando se carga una página, se crean una serie de objetos JavaScript asociados con los “principales” elementos HTML de la página:



# EJEMPLO



```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Página sencilla</title>
</head>
<body>
<p>Esta página es muy sencilla</p>
</body>
</html>
```





- Así, la siguiente etiqueta XHTML:  
<title>Página sencilla</title>
- Genera los siguientes dos nodos:



# ACCESO DIRECTO A LOS NODOS



- **getElementsByTagName()**

parrafos.html

```
var parrafos = document.getElementsByTagName("p");
```

Arreglo que  
contiene todos  
nodos  
seleccionados

Permite  
seleccionar  
determinados  
elementos

El elemento a  
seleccionar

De que parte se  
hace la selección



- **getElementByName()**
- Selecciona elementos cuyo nombre corresponda a la definición

formulario.html

```
<body>
Introduzca su nombre:
<FORM ACTION="recibir.php" METHOD="GET">
<INPUT TYPE="text" name="nombre" value="carlos">

<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
<script type="text/javascript">
var nom=document.getElementById("nombre");
alert (nom.value);
</script>
</body>
```



- **getElementById()**
- Selecciona elementos cuyo Id corresponda a la definición

enlace.html

```
<body>

<div id="cabecera">
Este es el enlace
</div>
<script type="text/javascript" language="javascript">
var cabecera =document.getElementById("cabecera");
alert(cabecera.innerHTML);
var enlace =document.getElementById("logo");
alert(enlace.innerHTML);
</script>
</body>
```



# ACCESO DIRECTO A LOS ATRIBUTOS XHTML



- Una vez que accedemos a un elemento es posible acceder a sus atributos, tanto de lectura como escritura

Atributo del elemento

```
<head>
<script type="text/javascript" language="javascript">
function changeSrc(){
alert(document.getElementById("myImage").src);
document.getElementById("myImage").src="logo2.jpg";
alert(document.getElementById("myImage").src);
}
</script>
```

# EVENTOS



- Señales generadas cuando ocurren acciones específicas
- Brindan la base para la interactividad
- Se disparan en el visualizador por acciones del usuario (principalmente)
- Se pueden construir acciones que reacciones a los eventos

# EVENTOS



- Ejemplos :
  - blur
    - Cuando el usuario hace click fuera de un campo en un formulario
  - click
    - Cuando el usuario hace click en un enlace o en un elemento de un formulario
  - change
    - Cuando el usuario cambia el valor de un campo
  - focus
    - Cuando se activa el foco en una entrada
  - load
    - Cuando se carga una página del navegador

# EVENTOS



- mouseover
  - Cuando el cursor (ratón) pasa por encima de una liga
- select
  - Cuando el usuario selecciona un campo de un elemento en un formulario
- submit
  - Cuando el usuario envía un formulario
- unload
  - Cuando un usuario abandona una página (para cerrar la ventana o cambiar de página)

# MANEJADORES COMO ATRIBUTOS DE LOS ELEMENTOS XHTML.



eventos1.html

```
<body onload="alert('La página se ha cargado completamente');">
<input type="button" value="Pinchame y verás" onclick="alert('Gracias por pinchar');" />
<div onmouseover="alert('Acabas de pasar el ratón por encima');">
Puedes pinchar sobre este elemento o simplemente pasar el ratón por encima
</div>
</body>
```

# MANEJADORES COMO FUNCIONES JAVASCRIPT EXTERNAS.



eventos2.html

```
function muestraMensaje() {
 alert('Gracias por pinchar');
}
```

```
<input type="button" value="Pinchame y verás" onclick="muestraMensaje()" />
```

# MANEJADORES "SEMÁNTICOS".



eventos3.html

```
<input id="pinchable" type="button" value="Pinchame y verás" />

<script type="text/javascript">
// Función externa
function muestraMensaje() {
alert('Gracias por pinchar');
}
// Asignar la función externa al elemento
document.getElementById("pinchable").onclick = muestraMensaje;
// Elemento XHTML
</script>
```

# EJEMPLO



imagen.html

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript" language="javascript">
function changeSrc(){
alert(document.getElementById("myImage").src);
document.getElementById("myImage").src="logo2.jpg";
alert(document.getElementById("myImage").src);
}
</script>
</head>
<body>

<input type="button" onclick="changeSrc()" value="cambiar
imagen">
<form name="formulario">
</body>
</html>
```

Gestor de Evento



# FORMULARIOS



- Cuando se carga una página web, navegador crea automáticamente un array llamado forms
- El arreglo contiene la referencia a todos los formularios de la página.

imagen.html

```
ingrese sus datos<form id="form1" name="form1" method="post" >
nombre <input name="txtTitulo" type="text" id="txtTitulo"/>

Apellido <input type="text" name="txtAutor" value="datos de prueba"/>

direccion<input type="text" name="txtISBN" />

Telefono <input type="text" name="txtEditorial" />

Email<input type="text" name="txtEdicion" />

<input type="submit" name="button" id="button" value="Enviar" />
</form>
```

```
<script type="text/javascript">
var f=document.forms;
var l=f.length;
alert('Existen '+l+' formularios en la pagina');
alert(f[0].txtAutor.value);
</script>
```

Arreglo de formularios

# MANEJAR RADIO BUTTONS



radio.html

```
<input type="radio" value="si" name="pregunta" id="pregunta_si"/> SI
<input type="radio" value="no" name="pregunta" id="pregunta_no"/> NO
<input type="radio" value="nsnc" name="pregunta" id="pregunta_nsnc"
checked="checked"/> NS/NC
<script type="text/javascript">
var elementos = document.getElementsByName("pregunta");
for(var i=0; i<elementos.length; i++) {
alert(" Elemento: " + elementos[i].value + "\n Seleccionado: " +
elementos[i].checked);
}
</script>
```

# LISTA DE SELECCION



lista.html

```
<select id="opciones" name="opciones">
<option value="1">Primer valor</option>
<option value="2">Segundo valor</option>
<option value="3">Tercer valor</option>
<option value="4">Cuarto valor</option>
</select>
<script language="javascript" type="text/javascript">
// Obtener la referencia a la lista
var lista = document.getElementById("opciones");
// Obtener el índice de la opción que se ha seleccionado
var indiceSeleccionado = lista.selectedIndex;
// Con el índice y el array "options", obtener la opción seleccionada
var opcionSeleccionada = lista.options[indiceSeleccionado];
// Obtener el valor y el texto de la opción seleccionada
var textoSeleccionado = opcionSeleccionada.text;
var valorSeleccionado = opcionSeleccionada.value;
alert("Opción seleccionada: " + textoSeleccionado + "\n Valor de la opción: " +
valorSeleccionado);
</script>
```

# VALIDACIÓN DE ENTRADA DE DATOS



validar.html

```
function validarForm()
{
 var formulario = document.getElementById("formulario");
 if(formulario.nombres.value.length==0) { //comprueba que no esté
vacío
 formulario.nombres.focus();
 alert('No has escrito tu nombre');
 return false; //devolvemos el foco
 }
 if(formulario.direccion.value.length==0) { //comprueba que no esté
vacío
 formulario.direccion.focus();
 alert('No has escrito tu direccion');
 return false; //devolvemos el foco
 }
 ...
 formulario.submit();
 return true;
}
```

# SOORTE DEL NAVEGADOR



- Los navegadores más modernos disponibles actualmente incluyen soporte de JavaScript hasta la tercera edición del estándar ECMA-262.
- Mayor diferencia reside en el *dialecto utilizado*
  - *Internet Explorer utiliza Jscript*
  - Resto de navegadores (Firefox, Opera, Chrome, Safari, Konqueror) utilizan JavaScript.



# FRAMEWORKS Y LIBRERIAS



- Aplicaciones web cada vez son más complejas
  - Incluyen efectos e interacciones avanzadas
- Programación de estas aplicaciones avanzadas se complica por varios motivos.
  - deben funcionar correctamente en los exploradores mas comunes
  - Es necesario aumentar la productividad de los desarrolladores
  - Entorno cada vez mas competitivo



han surgido librerías y *frameworks* específicos para el desarrollo de aplicaciones con JavaScript

Se reduce el tiempo de desarrollo

seguridad de que las aplicaciones funcionan igual de bien en cualquiera de los navegadores



# PROTOTYPE ([HTTP://WWW.PROTOTYPEJS.ORG/](http://www.prototypejs.org/))



- Es un framework que facilita el desarrollo de aplicaciones web con JavaScript y AJAX
- Se ha convertido en una referencia para AJAX y la base de muchos otros frameworks y librerías como script.aculo.us (<http://script.aculo.us/>)
  - `$()`. La *"función dólar"* es un atajo mejorado de la función `document.getElementById()`.

```
// Con JavaScript
var elemento =
document.getElementById('primero');
// Con Prototype
var elemento = $('primero');
```

# LIBRERÍA SCRIPTACULOUS



- Es una de las muchas librerías que han surgido para facilitar el desarrollo de aplicaciones JavaScript, permite:
  - **Efectos:** permite añadir de forma muy sencilla efectos especiales
  - **Controles:** define varios controles que se pueden añadir directamente a cualquier aplicación web

# JQUERY ([HTTP://JQUERY.COM/](http://jquery.com/))



- Es la librería JavaScript que ha irrumpido con más fuerza como alternativa a Prototype.
- Su versión comprimida apenas ocupa 20KB.
- **Funciones y métodos básicos**
  - *"función dolar": \$()*
  - separando expresiones con un carácter "," se puede seleccionar un número ilimitado de elementos.

```
// Selecciona todos los enlaces de la página
$('a')
// Selecciona el elemento cuyo id sea "primero"
$('#primero')
// Selecciona todos los h1 con class "titular"
$('h1.titular')
// Selecciona todo lo anterior
$('a, #primero, h1.titular')
```



```
// Selecciona todos los párrafos de la página que tengan al menos un enlace
$('p[a]')
// Selecciona todos los radiobutton de los formularios de la página
$('input:radio')
// Selecciona todos los enlaces que contengan la palabra "Imprimir"
$('a:contains("Imprimir")');
// Selecciona los div que no están ocultos
$('div:visible')
// Selecciona todos los elementos pares de una lista
$("ul#menuPrincipal li:even")
// Selecciona todos los elementos impares de una lista
$("ul#menuPrincipal li:odd")
// Selecciona los 5 primeros párrafos de la página
$("p:lt(5)")
Como se puede comprobar
```

# FUNCIONES PARA EVENTOS



- Ejecutar las instrucciones una vez que se ha cargado la página:

```
$(document).ready(function() {
```

```
...
```

```
});
```

```
// Establece la función manejadora del evento 'onclick'
```

```
// a todos los párrafos de la página
```

```
$('#p').click(function() {
```

```
 alert($(this).text());
```

```
});
```

```
// Establece la función manejadora del evento 'onblur'
```

```
// a los elementos de un formulario
```

```
$('#elFormulario :input').blur(function() {
```

```
 valida($(this));
```

```
});
```



```
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Documento sin título</title>
<script src="jquery-1.9.1.min.js" type="text/javascript"> </script>
</head>

<div id ="capa" style="padding:10x; background-color: #ff8800">Haz clic en un boton
</div>
<body>
<input type="button" value="Boton A" on onclick="$ (capa).html('Has heccho clic en
el boton A ')">
<input type="button" value="Boton B" on onclick="$ (capa).html('Has heccho clic en
el boton B ')">

</body>
```

# FUNCIONES PARA CSS



```
// Obtiene el valor de una propiedad CSS
// En este caso, solo para el primer 'div' de la página
$('div').css('background');
// Establece el valor de una propiedad CSS
// En este caso, para todos los 'div' de la página
$('div').css('color', '#000000');
// Establece varias propiedades CSS
// En este caso, para todos los 'div' de la página
$('div').css({ padding: '3px', color: '#CC0000' });
```

# OTRAS LIBRERIAS



- Dojo (<http://dojotoolkit.org/>) es mucho más que un framework,
- Mootools (<http://mootools.net/>) es un framework que destaca por su reducido tamaño y por lo modular de su desarrollo.
- Ext JS (<http://extjs.com/>) es otro de los frameworks más populares de JavaScript.



# LIBRERÍAS



- Yahoo! User Interface Library
- script.aculo.us: <http://script.aculo.us/>