Departamento de Lenguajes y Sistemas Informáticos

Universitat d'Alacant
Universidad de Alicante

JavaScript

Programación en Internet
Curso 2007-2008

Índice

Programación en Internet – Curso 2007-2008

Introducción

- JavaScript
- Aplicaciones
- Qué necesito para programar
- Versiones Netscape
- Versiones Microsoft
- ECMAScript

Programación en Internet – Curso 2007-2008

JavaScript

- Nombre original: Mocha → LiveScript (1995)
  - Netscape 2.0B3 (diciembre 1995)
- Lenguaje más estándar para script en Internet
- Cliente (Netscape Communicator) y servidor (Netscape Enterprise Server)
- JavaScript's LiveConnect →Java

Programación en Internet – Curso 2007-2008

JavaScript

- Lenguaje más estándar
- Indicado para programadores que provienen de C, C++ o Java
- No confundir con Java (Sun Microsystems)
- Microsoft: tiene su dialecto denominado Jscript (y JScript.NET)
- Dónde:
  - Cliente: Internet Explorer, Netscape Navigator, Opera, Mozilla, etc.
  - Servidor: ASP, Netscape Enterprise Server
  - Otros: PDF

Programación en Internet – Curso 2007-2008

Aplicaciones

- Durante muchos años era el único método para añadir dinamismo e interactividad a las páginas web
- En la actualidad existen otras alternativas, pero sigue siendo la tecnología más compatible

Programación en Internet – Curso 2007-2008

Aplicaciones

- JavaScript, DOM (*Document Object Model*) y BOM (*Browser Object Model*) son la combinación que permiten la programación en el cliente:
  - Validar entrada del usuario:
    - Reduce carga servidor
    - Reduce retrasos por errores usuario
    - Simplifica los programas
  - Proporcionar dinamismo, junto con DHTML u otras tecnologías

Programación en Internet – Curso 2007-2008

JavaScript y Applets de Java

JavaScript	Applets de Java
Interpretado	Compilado a bytecodes
Basado en objetos	Basado en clases
Código integrado	Objeto integrado
Tipos dinámicos No tipado	Tipos estáticos Fuertemente tipado

Programación en Internet – Curso 2007-2008

Versiones Netscape

Versión de JavaScript	Versión de Navigator
JavaScript 1.0	Navigator 2.0
JavaScript 1.1	Navigator 3.0
JavaScript 1.2	Navigator 4.0 – 4.05
JavaScript 1.3	Navigator 4.06 – 4.78
JavaScript 1.4	Ningún navegador
JavaScript 1.5	Navigator 6.x y Mozilla Application Suite, Firefox 1.0
JavaScript 1.6	Firefox 1.5
JavaScript 1.7	Firefox 2
JavaScript 2.0	*En desarrollo*

Programación en Internet – Curso 2007-2008

Versiones Microsoft

Host Application	1.0	2.0	3.0	4.0	5.0	5.1	5.5	5.6	.NET	8.0
Microsoft Internet Explorer 3.0	x									
Microsoft Internet Information Server 3.0		x								
Microsoft Internet Explorer 4.0			x							
Microsoft Internet Information Server 4.0			x							
Microsoft Internet Explorer 5.0					x					
Microsoft Internet Explorer 5.01						x				
Microsoft Windows 2000						x				
Microsoft Internet Explorer 5.5							x			
Microsoft Windows Millennium Edition							x			
Microsoft Internet Explorer 6.0								x		
Microsoft Windows XP								x		
Microsoft Windows Server 2003								x		
Microsoft .NET Framework 1.0									x	

ECMAScript

- Ecma International → ECMA 262
  - Ed. 1: junio 1997
  - Ed. 2: junio 1998
  - Ed. 3: diciembre 1999
- JavaScript y JScript son una implementación de ECMAScript, pero proporcionan características adicionales

Qué necesito para programar

- Editor ASCII estándar → Syntax highlight
- Navegador
- Ficheros: .js

Programación en Internet – Curso 2007-2008

El lenguaje (I)

- Características básicas
- Comentarios
- Declaración de variables
- Ámbito de las variables
- Caracteres especiales
- Operadores
- Palabras reservadas

Programación en Internet – Curso 2007-2008

El lenguaje (II)

- Sentencias condicionales
- Sentencias de repetición
- Sentencias de manipulación de objetos
- Declaración de funciones
- Funciones predefinidas

Programación en Internet – Curso 2007-2008

El lenguaje (y III)

- Tratamiento de cadenas
- Operaciones matemáticas

Programación en Internet – Curso 2007-2008

Características básicas

- Sintaxis prácticamente idéntica a Java y C++
- Sensible a minúsculas / mayúsculas (*case sensitive*)
- El punto y coma (;) → No obligatorio
- Bloque de código: {...}

Comentarios

- No afectan a la velocidad de ejecución
- Dos estilos:
  - // una sola línea
  - /* un comentario
    que ocupa varias líneas */

Programación en Internet – Curso 2007-2008

Declaración de variables (I)

- No es necesario declararlas. Se emplea la palabra reservada **var**
- Normas:
  - Letras, números o _
  - Primer carácter: no número
  - Variable ≠ Palabra reservada
- Valor inicial → **null**

Programación en Internet – Curso 2007-2008

Declaración de variables (y II)

- Lenguaje sin tipos
- Tipos de una variable:
  - Cadenas (‘’ y “”)
    - Para que una cadena se extienda múltiples líneas, cada línea tiene que terminar con el carácter de escape \
  - Números enteros y en coma flotante
  - Booleanos → true y false
  - Nulo → null

Programación en Internet – Curso 2007-2008

Ámbito de las variables

- Variable global / local
- Se puede crear una variable local con el mismo nombre que una global gracias a **var**
- Distintos ámbitos locales entre distintas funciones

Programación en Internet – Curso 2007-2008

Caracteres especiales

Carácter	Significado
\b	Retroceso (*backspace*)
\f	Salto de página (*form feed*)
\n	Salto de línea (*new line*)
\r	Retorno de carro (*carriage return*)
\t	Tabulador
\'	Apóstrofe o comilla simple
\"	Comilla doble
\\	Barra invertida (*baskslash)*
\XXX	El carácter de la codificación Latin-1 especificado por los tres dígitos octales entre 0 y 377.
\xXX	El carácter de la codificación Latin-1 especificado por los dos dígitos hexadecimales entre 00 y FF.
\uXXXX	El carácter Unicode especificado por los cuatro dígitos hexadecimales entre 0000 y FFFF.

Programación en Internet – Curso 2007-2008

Operadores

Precedencia de los operadores	
Tipo de operador	**Operador**
Coma	,
Asignación	= += -= *= /= %= <<= >>= >>>= &= ^= |=
Condicional	?:
O lógico (OR)	||
Y lógico (AND)	&&
O bit a bit (OR)	|
O exclusiva bit a bit (XOR)	^
Y bit a bit (AND)	&
Igualdad	== != === !==
Relacional	< <= > >=
Desplazamiento bit a bit	<< >> >>>
Suma y resta	+ -
Multiplicación y división	* / %
Negación e incremento	! ~ - + ++ -- typeof void delete
Llamada a función	()
Creación de instancias	new
Miembro	. []

Programación en Internet – Curso 2007-2008

Palabras reservadas

Palabras reservadas de JavaScript			
abstract	else	instanceof	switch
boolean	enum	int	synchronized
break	export	interface	this
byte	extends	long	throw
case	false	native	throws
catch	final	new	transient
char	finally	null	true
class	float	package	try
const	for	private	typeof
continue	function	protected	var
debugger	goto	public	void
default	if	return	volatile
delete	implements	short	while
do	import	static	with
double	in	super	

Programación en Internet – Curso 2007-2008

Sentencias condicionales

```
if(condicion) {
  sen1
}
[else {
  sen2
}]
```

```
switch(expresion){
   case et1 :
      sen1;
      [break;]
   case et2 :
      sen2;
      [break;]
   ...
   [default :
      sen;]
}
```

Programación en Internet – Curso 2007-2008

Sentencias de repetición

```
for ([expInicial]; [condicion]; [expIncremento]) {
   sentencias
}
```

```
do {
  sentencias
} while(condicion)
```

```
while(condicion)
{
  sentencias
}
```

break → Finaliza un bucle

continue → Pasa a la siguiente iteración

Programación en Internet – Curso 2007-2008

Sentencias de manipulación de objetos

```
for(variable in objeto)
{
  sentencias
}
```

```
with(obj)
{
  sentencias
}
```

Programación en Internet – Curso 2007-2008

Declaración de funciones (I)

```
function nombre([arg1 [, arg2 [, …]]]) {
  sentencias
}
```

Para devolver un valor o detener la ejecución → **return**

Programación en Internet – Curso 2007-2008

Declaración de funciones (II)

- Paso de argumentos:
  - Los tipos primitivos se pasan por valor (se pasa una copia del valor)
  - Los arrays y objetos se pasan por referencia: cualquier cambio afecta al original

Programación en Internet – Curso 2007-2008

Funciones predefinidas

- eval
- isFinite e isNaN
- parseInt y parseFloat
- Number y String
- escape y unescape

Programación en Internet – Curso 2007-2008

Objetos (I)

- Lenguaje basado en objetos, pero no orientado a objetos → No hay clases, herencia, visibilidad, etc.
- Sentencias: **for (... in ...)** y **with()**
- Operador "." o "[ ]" (arrays asociativos)

```
window.status = "Bienvenido a JavaScript";
window.alert("2 + 2 = " + (2 + 2));

window["status"] = "Bienvenido a
  JavaScript";
window["alert"]("2 + 2 = " + (2 + 2));
```

Programación en Internet – Curso 2007-2008

Objetos (II)

- Creación de objetos:
  - Inicializadores de objetos:

```
objeto = {prop1:val1, ..., propN:valN};
```

  - Funciones constructoras:

```
function ObjConstructor(arg1, ..., argN)
{
  this.prop1 = arg1; ...; this.propN = argN;
}

objeto = new ObjConstructor(val1, ..., valN);
```

Programación en Internet – Curso 2007-2008

Objetos (y III)

- Métodos de un objeto:
  - Asignad a una propiedad del objeto el nombre de una función
  - Emplead en la función **this**
- Eliminación de objetos → **delete**

Programación en Internet – Curso 2007-2008

Tratamiento de cadenas (I)

- Cadenas en JavaScript:
  - Cadenas literales (‘’ y “”)
  - Objeto **String**
- JavaScript convierte automáticamente las cadenas literales a objetos **String**
- Propiedades: **length**
- Muy útil para validar los datos de los formularios HTML

Programación en Internet – Curso 2007-2008

Tratamiento de cadenas (y II)

- Métodos:
  - `charAt(indice): 0 ... length - 1`
  - `concat(cad1, cad2, ..., cadn)` → ”+”
  - `fromCharCode(num1, num2, …)`
  - `indexOf(valor, inicio)`
  - `lastIndexOf(valor, inicio)`
  - `slice(inicio, fin)`
  - `split(separador, limite)`
  - `...`

Programación en Internet – Curso 2007-2008

Operaciones matemáticas

- Objeto Math
- Propiedades: `E, LN10, LN2, LOG10E, LOG2E, PI, ...`
- Métodos: `abs(), acos(), asin(), atan(), ceil(), cos(), sin(), tan(), exp(), floor(), max(), min(), random(), ...`
- Útiles para hacer cálculos en la parte cliente

Programación en Internet – Curso 2007-2008

Validación de formularios

- DOM
- Validación cliente / servidor
- Validación campo nulo
- Validación alfabética
- Validación numérica
- Validación de una fecha

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Posibles escenarios:
  - Usuario con un navegador sin soporte para JavaScript (por ejemplo, usuario discapacitado)
  - Usuario que ha desactivado el soporte de JavaScript (por ejemplo, por razones de serguridad)
  - Buscador (pocos entienden JavaScript)

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Problema típico: ventanas de popup

JavaScript:

```
function popUp(winURL) {
window.open(winURL,"popup","width=320,height=480");
}
```

HTML:

```
<a href="javascript:popUp('http://www.ua.es/');">UA</a>

<a href="" onclick="popUp('http://www.ua.es/');">UA<a>
```

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Solución: *graceful degradation*
  - Cuando un sistema “se degrada con gracia”, su funcionalidad se puede ver reducida, pero no falla completamente

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Solución: añadir una URL en el enlace

```
<a href="http://www.ua.es/"
  onclick="popUp('http://www.ua.es'; return
  false;">UA</a>
```

- `return false` impide que el enlace (`href`) sea seguido
- La funcionalidad se ve reducida porque no se abre en una nueva ventana, pero no falla completamente

JavaScript no intrusivo

- Mejora: evita repetir la URL (inconsistencia) y ahorra espacio

```
<a href="http://www.ua.es/"
  onclick="popUp(this.getAttribute('href'));
  return false;">UA</a>

<a href="http://www.ua.es/"
  onclick="popUp(this.href); return
  false;">UA</a>
```

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Con la solución anterior, el código JavaScript está escrito directamente en las etiquetas
- Problemas:
  - Dificulta la escritura
  - Dificulta la lectura
  - Dificulta el mantenimiento
- ¿No se puede separar como con CSS?

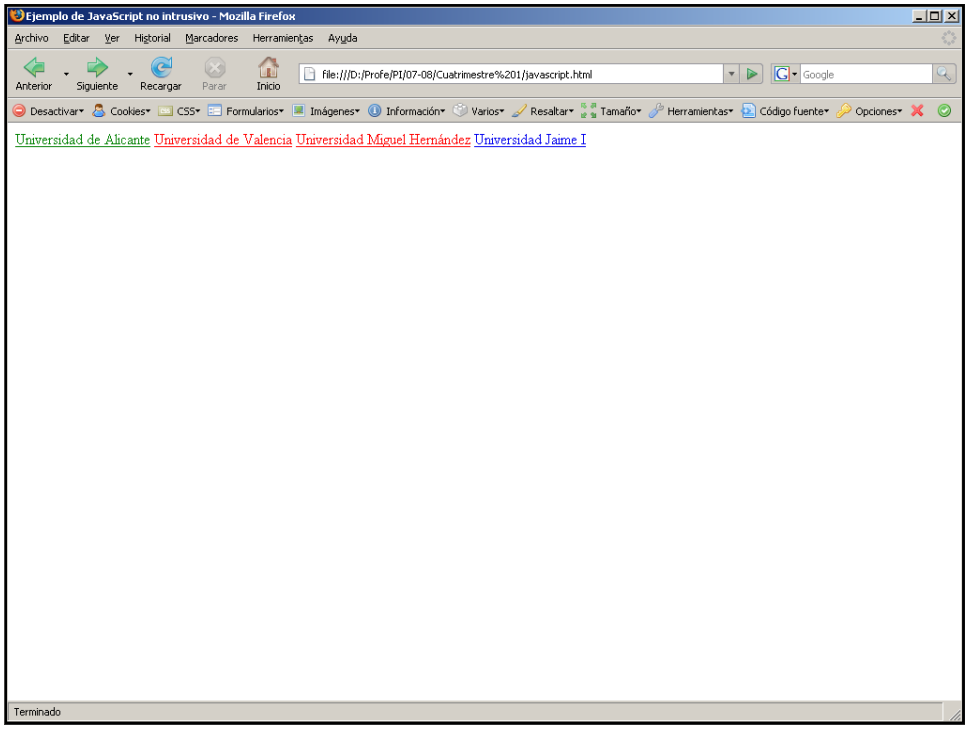
JavaScript no intrusivo

- Una página web se puede construir mediante capas:
  - Capa de contenido → Contenido y marcado (XHTML)
  - Capa de presentación → CSS
  - Capa de comportamiento → JavaScript
- Una página web construida de esta forma (*progressive enhancement*) casi siempre es *graceful degradation*

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

- Se puede asignar código a un manejador de eventos desde JavaScript:
  `element.event = action;`
- Seleccionar un elemento:
  - `getElementById(id).event = action;`
  - Usar una combinación de `getElementsByTagName` y `getAttribute` para asignar código a elementos con un atributo específico



Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

```
<html>
<head>
<title>Ejemplo de JavaScript no intrusivo</title>
<style>
.bien {color: green;}

.mal {color: red;}
</style>
```

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

```
<script type="text/javascript">
window.onload = prepareLinks;
function prepareLinks() {
  var links = document.getElementsByTagName("a");
  for (var i=0; i<links.length; i++) {
    if (links[i].getAttribute("class").indexOf("popup") >= 0) {
      links[i].onclick = function() {
        popUp(this.getAttribute("href"));
        return false;
      }
    }
  }
}
function popUp(winURL) {
  window.open(winURL,"popup","width=320,height=480");
}
</script></head>
```

Programación en Internet – Curso 2007-2008

JavaScript no intrusivo

```
<body>
<p>
<a href="http://www.ua.es/" class="bien
   popup">Universidad de Alicante</a>
<a href="http://www.uv.es/" class="mal
   popup">Universidad de Valencia</a>
<a href="http://www.umh.es/" class="mal
   popup">Universidad Miguel Hernández</a>
<a href="http://www.uji.es/" class="popup">Universidad
   Jaime I</a>
</p>
</body>
</html>
```

Compatibilidad hacia atrás

- Problema: existe una gran diversidad de grados de soporte de JavaScript, del BOM y del DOM
- ¿Cómo asegurar que el código JavaScript no será mal interpretado o no funcionará por no existir algún elemento?

Programación en Internet – Curso 2007-2008

Compatibilidad hacia atrás

- Solución: técnica de **detección de objetos**
  - Comprobar si existe la función u objeto que necesitamos
  - Un método se trata como una función

Compatibilidad hacia atrás

```
function prepareLinks() {
  if (!document.getElementsByTagName) return false;

  var links = document.getElementsByTagName("a");

  for (var i=0; i<links.length; i++) {
    if (links[i].getAttribute("class").indexOf("popup")
 >= 0) {
      links[i].onclick = function() {
        popUp(this.getAttribute("href"));
        return false;
      }
    }
  }
}
```


Compatibilidad hacia atrás

- Desaconsejado: No es una buena técnica *browser sniffing*: obtener información del navegador para averiguar su nombre y versión
- Problemas:
  - Algunos navegadores mienten
  - Algunos navegadores (Opera) permiten al usuario modificar estos valores
  - Con la cantidad actual de navegadores es un problema muy complejo (muchas combinaciones)
  - Nuevas versiones cada poco tiempo

Programación en Internet – Curso 2007-2008

Compatibilidad hacia atrás

- Ejemplo browser sniffing:

```
<script type="text/javascript">
var agt = navigator.userAgent.toLowerCase();
var major = parseInt(navigator.appVersion);
var minor = parseFloat(navigator.appVersion);

var is_nav  = ((agt.indexOf('mozilla')!= -1) && ((agt.indexOf('spoofer') ==
   -1) && (agt.indexOf('compatible') == -1)));
var is_ie = ((agt.indexOf("msie") != -1) && (agt.indexOf("opera") == -1));
var is_opera = (agt.indexOf("opera") != -1);

if(is_nav)
  document.write("Navigator");
else if(is_ie)
  document.write("Internet Explorer");
else if(is_opera)
  document.write("Opera");
</script>
```