



Departamento de Lenguajes y  
Sistemas Informáticos



Universitat d'Alacant  
Universidad de Alicante

# JavaScript

Programación en Internet  
Curso 2005-2006

Programación en Internet – Curso 2005-2006

## Contenidos

- Introducción
- El lenguaje
- Validación de formularios

Programación en Internet – Curso 2005-2006

## Introducción

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

Programación en Internet – Curso 2005-2006

## JavaScript

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

## 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
- Dónde:
  - Cliente: Internet Explorer, Netscape Navigator, Opera, Mozilla, etc.
  - Servidor: ASP, Netscape Enterprise Server

## Aplicaciones

- Validar entrada del usuario:
  - Reduce carga servidor
  - Reduce retrasos por errores usuario
  - Simplifica los programas
- Proporcionar dinamismo, junto con DHTML u otras tecnologías

## Qué necesito para programar

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

## 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 2005-2006

## 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
<i>JavaScript 2.0</i>	<i>En desarrollo</i>

Programación en Internet – Curso 2005-2006

## Versiones Microsoft

Versión de JScript	Versión de Microsoft
JScript v1	IE 3
JScript v2	IIS 3.0
JScript v3	IE 4
JScript v4	Visual Interdev 6.0
JScript v5	Windows 2000, Office 2000, IE 5 y 6

## El lenguaje (I)

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

## El lenguaje (II)

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

Programación en Internet – Curso 2005-2006

## El lenguaje (y III)

- Tratamiento de cadenas
- Operaciones matemáticas

Programación en Internet – Curso 2005-2006

## 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 \*/

## Declaración de variables (I)

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



## 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

## 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 2005-2006

## Caracteres especiales

Carácter	Significado
\b	Retroceso ( <i>backspace</i> )
\f	Salto de página ( <i>form feed</i> )
\n	Salto de línea ( <i>new line</i> )
\r	Retorno de carro ( <i>carriage return</i> )
\t	Tabulador
\'	Apóstrofe o comilla simple
\"	Comilla doble
\\	Barra invertida ( <i>backslash</i> )
\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 2005-2006

## 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	. []

## 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	

## Sentencias condicionales

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

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

## 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

## Sentencias de manipulación de objetos

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

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

## Declaración de funciones (I)

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

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

## 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

## Funciones predefinidas

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

## 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));
```

## 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);
```

## 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**

## 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

## 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)
  - ...



## 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

## Validación de formularios

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