Appendix C. JSLint

# Apéndice C. JSLint

¿Qué error extravía nuestros ojos y oídos?

-WilliamShakespeare, *The Comedy of Errors*

Cuando C era un lenguaje de programación joven, había varios errores de programación comunes que los compiladores primitivos no detectaban, por lo que se desarrolló un programa accesorio llamado lint que escaneaba un archivo fuente en busca de problemas.

A medida que C maduró, la definición del lenguaje se reforzó para eliminar algunas inseguridades, y los compiladores mejoraron a la hora de emitir advertencias. lint ya no es necesario.

JavaScript es un lenguaje joven para su edad. Originalmente estaba pensado para realizar pequeñas tareas en páginas web, tareas para las que Java era demasiado pesado y torpe. Pero JavaScript es un lenguaje muy capaz, y ahora se utiliza en proyectos de mayor envergadura. Muchas de las características que se pensaron para facilitar el uso del lenguaje son problemáticas para proyectos más grandes, por lo que se necesita un lint para JavaScript: JSLint, un comprobador y verificador de sintaxis de JavaScript.

JSLint es una herramienta de calidad de código para JavaScript. Toma un texto fuente y lo escanea. Si encuentra un problema, devuelve un mensaje describiendo el problema y una ubicación aproximada dentro del código fuente. El problema no es necesariamente un error de sintaxis, aunque a menudo lo es. JSLint examina algunas convenciones de estilo, así como problemas estructurales. No prueba que tu programa sea correcto. Sólo proporciona otro par de ojos para ayudar a detectar problemas.

JSLint define un subconjunto profesional de JavaScript, un lenguaje más estricto que el definido por la tercera edición de la norma *ECMAScript Language Specification*. El subconjunto está estrechamente relacionado con las recomendaciones de estilo del [Capítulo 9](ch09.html).

JavaScript es un lenguaje descuidado, pero dentro de él hay un lenguaje elegante y mejor. JSLint te ayuda a programar en ese lenguaje mejor y a evitar la mayor parte de la bazofia.

Puedes encontrar JSLint en <http://www.JSLint.com/>.

# Variables y funciones indefinidas

El mayor problema de JavaScript es su dependencia de las variables globales, sobre todo de las variables globales implícitas. Si una variable no se declara explícitamente (normalmente con la declaración var ), JavaScript asume que la variable es global. Esto puede ocultar nombres mal escritos y otros problemas.

JSLint espera que todas las variables y funciones se declaren antes de ser utilizadas o invocadas. Esto le permite detectar variables globales implícitas. También es una buena práctica porque facilita la lectura de los programas.

A veces un archivo depende de variables y funciones globales que están definidas en otra parte. Puedes identificarlos ante JSLint incluyendo un comentario en tu archivo que enumere las funciones y objetos globales de los que depende tu programa, pero que no están definidos en tu programa o archivo de script.

Un comentario de declaración global puede utilizarse para enumerar todos los nombres que utilizas intencionadamente como variables globales. JSLint puede utilizar esta información para identificar errores ortográficos y declaraciones olvidadas en var. Una declaración global puede tener este aspecto

/\*global getElementByAttribute, breakCycles, hanoi \*/

Una declaración global comienza con /\*global. Observa que no hay espacio antes de g. Puedes tener tantos comentarios /\*global como quieras. Deben aparecer antes del uso de las variables que especifican.

Algunas globales pueden predefinirse para ti (véase la sección posterior["Opciones](apcs03.html)"). Selecciona la opción "Asumir un navegador" (browser) para predefinir las propiedades globales estándar que proporcionan los navegadores web, como window y document y alert. Selecciona la opción "Asumir Rhino" (rhino) para predefinir las propiedades globales que proporciona el entorno Rhino. Selecciona la opción "Asumir un Widget Yahoo" (widget) para predefinir las propiedades globales proporcionadas por el entorno Yahoo! Widgets.