Chapter 7. Regular Expressions

# Capítulo 7. Expresiones regulares Expresiones regulares

Mientras que lo contrario trae la dicha, Y es un modelo de paz celestial. A quién debemos emparejar con Enrique, siendo rey . . .

-WilliamShakespeare, *The First Part of Henry the Sixth*

Muchas de las características de JavaScript se tomaron prestadas de otros lenguajes. La sintaxis procede de Java, las funciones de Scheme, la herencia prototípica de Self y la función de Expresión Regular de JavaScript de Perl.

A *regular expression* es la especificación de la sintaxis de un lenguaje sencillo. Las expresiones regulares se utilizan con métodos para buscar, sustituir y extraer información de cadenas. Los métodos que funcionan con expresiones regulares son regexp.exec, regexp.test, string.match, string.replace, string.search y string.split. Todos ellos se describirán en el [Capítulo 8](ch08.html). Las expresiones regulares suelen tener una ventaja de rendimiento significativa sobre las operaciones de cadena equivalentes en JavaScript.

Las expresiones regulares proceden del estudio matemático de los lenguajes formales. Ken Thompson adaptó el trabajo teórico de Stephen Kleene sobre los lenguajes de tipo 3 a un comparador de patrones práctico que pudiera incorporarse a herramientas como editores de texto y lenguajes de programación.

La sintaxis de las expresiones regulares en JavaScript se ajusta estrechamente a las formulaciones originales de los Laboratorios Bell, con algunas reinterpretaciones y extensiones adoptadas de Perl. Las reglas para escribir expresiones regulares pueden ser sorprendentemente complejas porque interpretan caracteres en algunas posiciones como operadores, y en posiciones ligeramente diferentes como literales. Peor que ser difíciles de escribir, esto hace que las expresiones regulares sean difíciles de leer y peligrosas de modificar. Es necesario tener un conocimiento bastante completo de toda la complejidad de las expresiones regulares para leerlas correctamente. Para mitigar esto, he simplificado un poco las reglas. Tal como se presentan aquí, las expresiones regulares serán ligeramente menos concisas, pero también serán ligeramente más fáciles de utilizar correctamente. Y eso es bueno, porque las expresiones regulares pueden ser muy difíciles de mantener y depurar.

Las expresiones regulares actuales no son estrictamente regulares, pero pueden ser muy útiles. Las expresiones regulares tienden a ser extremadamente escuetas, incluso crípticas. Son fáciles de usar en su forma más simple, pero pueden volverse rápidamente desconcertantes. Las expresiones regulares de JavaScript son difíciles de leer en parte porque no permiten comentarios ni espacios en blanco. Todas las partes de una expresión regular están muy juntas, lo que las hace casi indescifrables. Esto es especialmente preocupante cuando se utilizan en aplicaciones de seguridad para escanear y validar. Si no puedes leer y comprender una expresión regular, ¿cómo puedes confiar en que funcionará correctamente para todas las entradas? Sin embargo, a pesar de sus evidentes inconvenientes, las expresiones regulares se utilizan ampliamente.

[Subtema 1 de 2: (Ver todo)](ch07.html)[siguiente](ch07_2.html)