Chapter 1. Good Parts

# Capítulo 1. Las partes buenas

...dejando a un lado los atractivos de mis partes buenas, no tengo otros encantos.

-WilliamShakespeare, *The Merry Wives of Windsor*

Cuando era un joven programador novato, me aprendía todas las características de los lenguajes que utilizaba e intentaba utilizarlas todas cuando escribía. Supongo que era una forma de presumir, y supongo que funcionaba porque yo era la persona a la que acudías si querías saber cómo utilizar una función concreta.

Con el tiempo me di cuenta de que algunas de esas funciones eran más problemáticas de lo que valían. Algunas estaban mal especificadas, por lo que era más probable que causaran problemas de portabilidad. Algunas daban lugar a código difícil de leer o modificar. Algunas me indujeron a escribir de un modo demasiado complicado y propenso a errores. Y algunas de esas características eran errores de diseño. A veces, los diseñadores de lenguajes cometen errores.

La mayoría de los lenguajes de programación contienen partes buenas y partes malas. Descubrí que podía ser mejor programador utilizando sólo las partes buenas y evitando las malas. Después de todo, ¿cómo puedes construir algo bueno con partes malas?

Rara vez es posible que los comités de normalización eliminen las imperfecciones de un lenguaje, porque hacerlo provocaría la rotura de todos los programas malos que dependen de esas partes malas. Normalmente no pueden hacer nada, salvo amontonar más funciones sobre el montón de imperfecciones existentes. Y las nuevas características no siempre interactúan armoniosamente, produciendo así más partes malas.

Pero *you* tienes el poder de definir tu propio subconjunto. Puedes escribir mejores programas basándote exclusivamente en las partes buenas.

JavaScript es un lenguaje con más que su parte de partes malas. Pasó de la inexistencia a la adopción global en un periodo de tiempo alarmantemente corto. Nunca tuvo un intervalo en el laboratorio en el que pudiera probarse y pulirse. Entró directamente en Netscape Navigator 2 tal cual, y fue muy duro. Cuando los applets Java™ fracasaron, JavaScript se convirtió en el "Lenguaje de la Web" por defecto. La popularidad de JavaScript es casi totalmente independiente de sus cualidades como lenguaje de programación.

Afortunadamente, JavaScript tiene algunas partes extraordinariamente buenas. En JavaScript hay un lenguaje bello, elegante y muy expresivo que está enterrado bajo un montón humeante de buenas intenciones y meteduras de pata. La mejor naturaleza de JavaScript está tan eficazmente oculta que durante muchos años la opinión predominante sobre JavaScript fue que era un juguete antiestético e incompetente. Mi intención aquí es exponer las bondades de JavaScript, un lenguaje de programación extraordinario y dinámico. JavaScript es un bloque de mármol, y yo desmenuzo las características que no son bellas hasta que se revela la verdadera naturaleza del lenguaje. Creo que el elegante subconjunto que he tallado es muy superior al lenguaje en su conjunto, ya que es más fiable, legible y fácil de mantener.

Este libro no intentará describir completamente el lenguaje. En su lugar, se centrará en las partes buenas con advertencias ocasionales para evitar las malas. El subconjunto que se describirá aquí puede utilizarse para construir programas fiables y legibles, grandes y pequeños. Centrándonos sólo en las partes buenas, podemos reducir el tiempo de aprendizaje, aumentar la robustez y salvar algunos árboles.

Quizá la mayor ventaja de estudiar las partes buenas es que puedes evitar la necesidad de desaprender las partes malas. Desaprender los patrones malos es muy difícil. Es una tarea dolorosa a la que la mayoría de nosotros nos enfrentamos con extrema reticencia. A veces las lenguas se subconfiguran para que funcionen mejor para los estudiantes. Pero en este caso, estoy subconfigurando JavaScript para que funcione mejor para los profesionales.

[Subtema 1 de 2: (Ver todo)](ch01.html)[siguiente](ch01_2.html)