

linguagem de programação. E JavaScript há muito tempo superou suas raízes em linguagem de script para se tornar um general robusto e eficiente. Linguagem de propósito adequada para engenharia e projetos graves de software com enormes bases de código.

JavaScript: nomes, versões e modos

JavaScript foi criado no Netscape nos primeiros dias da web e tecnicamente, "JavaScript" é uma marca registrada licenciada da Sun Microsystems (agora Oracle) usada para descrever o Netscape's (agora Mozilla's)

implementação do idioma. Netscape enviou o idioma para padronização à ECMA - a Associação Europeia de Fabricante de Computadores - e por questões de marca registrada, o padronizado. A versão do idioma ficou presa com o nome desajeitado "Ecmascript". Na prática, todos apenas chamam o idioma JavaScript. Este livro usa o nome "ecmascript" e a abreviação "es" para consultar o padrão de idioma e as versões desse padrão.

Na maior parte dos 2010, a versão 5 do padrão Ecmascript foi suportada por toda a web navegadores. Este livro trata o ES5 como a linha de base da compatibilidade e não discute mais versões anteriores do idioma. O ES6 foi lançado em 2015 e adicionou novos recursos - incluindo uma sintaxe de módulo - que alterou o JavaScript de uma linguagem de script em um grave, de uso geral idioma adequado para engenharia de software em larga escala. Desde o ES6, a especificação do ECMAScript tem mudado - se para uma cadência de liberação anual e versões do idioma - ES2016, ES2017, ES2018, ES2019 e ES2020 - agora são identificados por ano de lançamento.

À medida que o JavaScript evoluiu, os designers de idiomas tentaram corrigir falhas no início (pré-ES5) versões. Para manter a compatibilidade com versões anteriores, não é possível remover recursos legados, não importa o quão falho. Mas no ES5 e mais tarde, os programas podem optar pelo modo estrito de JavaScript no qual um

Número de erros de idioma inicial foram corrigidos. O mecanismo de opção é o ? Use

Diretiva estrita ? descrita em §5.6.3. Essa seção também resume as diferenças entre o legado

JavaScript e JavaScript rigoroso. No ES6 e mais tarde, o uso de novos recursos de linguagem frequentemente implicitamente

invoca o modo rigoroso. Por exemplo, se você usar a palavra-chave da classe ES6 ou criar um módulo ES6, então todo o código dentro da classe ou módulo é automaticamente rigoroso, e os recursos antigos e defeituosos não são

disponíveis nesses contextos. Este livro abordará os recursos legados do JavaScript, mas é cuidadoso para apontar que eles não estão disponíveis no modo rigoroso.

Para ser útil, todo idioma deve ter uma plataforma ou biblioteca padrão,

para realizar coisas como entrada e saída básicas. O principal javascript

A linguagem define uma API mínima para trabalhar com números, texto,

Matrizes, conjuntos, mapas e assim por diante, mas não inclui nenhuma entrada ou saída

funcionalidade. Entrada e saída (bem como recursos mais sofisticados, como networking, armazenamento e gráficos) são de responsabilidade do