

Reflection:

This volume was a summary of the topics covered in the series ahead.

The author is insistent, and rightfully so, that learning Js in depth is a requirement for being a Js developer. Apparently too many developers, (I have been guilty of this), learn the bare-minimum to get by on a feature or framework, without true understanding.

The future of Js is foreseeable on the horizon and it is built on the fundamentals that came before. Like life, Js keeps changing
Dive on in!

Scope & Closure:

Js is not an interpreted language. Understand how the Js engine compiles code right before and sometimes during execution. Find and deal with variable and function declarations, hoisting.

Understand lexical scope will lend itself to closure, and this the module pattern of code organization.

this & Object Prototypes:

`this`, keyword value is dynamically bound based on the context, or how, a given function is executed.

Object prototypes discussed by author taking pains to label any classical inheritance behaviors in quotes to denote the quasi- status. Author prefers “behavior delegation”, as the correct design pattern.

Types & Grammar:

Deep dive into coercion and understanding the subtleties.

Async & Performance:

Understanding terminology of asynchronous programming. I suspect event loop shall be discussed.

Using promises and generators.

Optimizing Js to with program parallelism and other techniques.

ES6 & Beyond:

"New" syntax, new data structures new API, new processing capabilities.

A look ahead at the landscape and evolution of Js.

To ES6 & beyond!