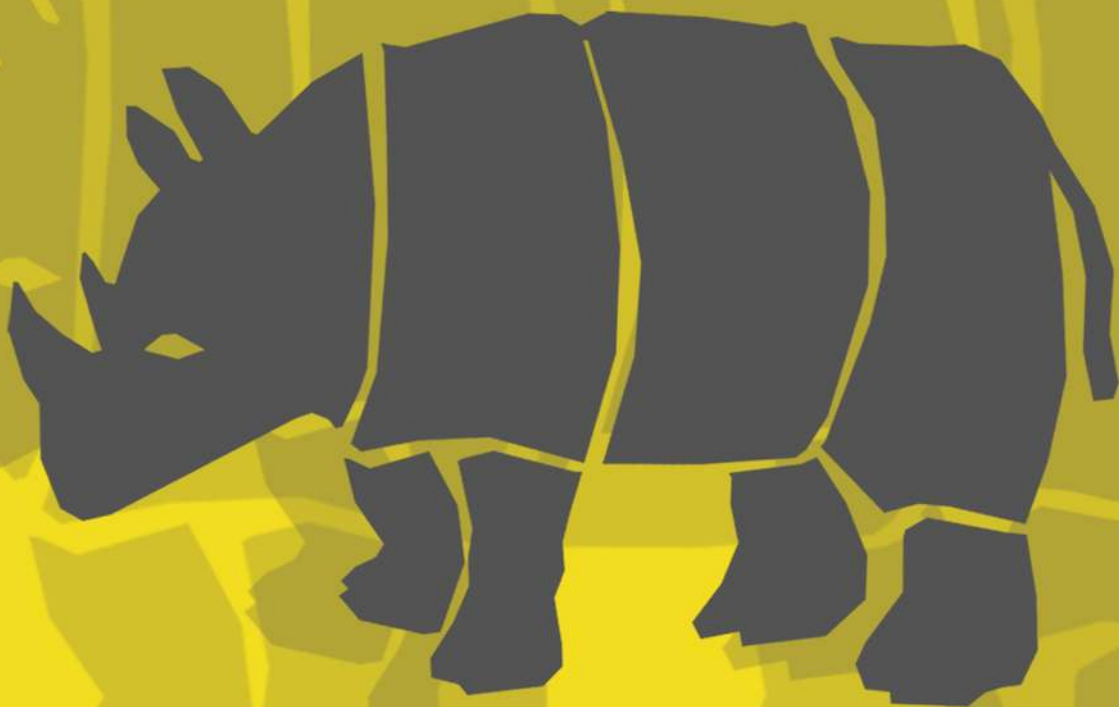


JavaScript

FOR IMPATIENT PROGRAMMERS



Dr. Axel Rauschmayer

Contents

I	Background	7
1	About this book	9
1.1	What's in this book?	9
1.2	What is not covered by this book?	9
1.3	This book isn't finished, yet	9
1.4	Can I buy a print edition?	10
1.5	Will there be a free online version?	10
1.6	Acknowledgements	10
2	Tips for reading this book	11
2.1	What are the "advanced" chapters and sections about?	11
2.2	What should I read if I'm <i>really</i> impatient?	11
2.3	How do I submit feedback and corrections?	11
2.4	I'm occasionally seeing type annotations – how do those work?	11
3	History and evolution of JavaScript	13
3.1	How JavaScript was created	13
3.2	Standardization	13
3.3	Evolving JavaScript: don't break the web	16
4	FAQ: JavaScript	17
4.1	Why does JavaScript fail silently so often?	17
II	First steps	19
5	The big picture	21
5.1	What are you learning in this book?	21
5.2	The structure of browsers and Node.js	21
5.3	Trying out JavaScript code	22
5.4	Further reading	25
6	Syntax	27
6.1	An overview of JavaScript's syntax	27
6.2	(Advanced)	31
6.3	Identifiers	31
6.4	Statement vs. expression	32
6.5	Syntactically ambiguous constructs	33

6.6	Semicolons	35
6.7	Automatic semicolon insertion (ASI)	36
6.8	Semicolons: best practices	38
6.9	Strict mode	39
7	Assertion API	41
7.1	Assertions in software development	41
7.2	How assertions are used in this book	41
7.3	Normal comparison versus deep comparison	42
7.4	Quick reference: module assert	42
8	Getting started with quizzes and exercises	45
8.1	Quizzes	45
8.2	Exercises	45
8.3	Unit tests in JavaScript	46
III	Variables and values	49
9	Variables and assignment	51
9.1	let	51
9.2	const	51
9.3	Deciding between let and const	52
9.4	Variables are block-scoped	53
10	Values	55
10.1	What's a type?	55
10.2	JavaScript's type hierarchy	55
10.3	The types of the language specification	55
10.4	Primitive values versus objects	56
10.5	Classes and constructor functions	58
10.6	Constructor functions associated with primitive types	59
10.7	The operators typeof and instanceof: what's the type of a value?	59
10.8	Converting between types	61
11	Operators	63
11.1	Two important rules for operators	63
11.2	The plus operator (+)	64
11.3	Assignment operators	64
11.4	Equality: == versus ===	65
11.5	Ordering operators	67
11.6	Various other operators	68
IV	Primitive values	69
12	The non-values undefined and null	71
12.1	undefined vs. null	71
12.2	Occurrences of undefined and null	72
12.3	Checking for undefined or null	72
12.4	undefined and null don't have properties	73

13 Booleans	75
13.1 Converting to boolean	75
13.2 Falsy and truthy values	76
13.3 Conditional operator (?:)	78
13.4 Binary logical operators: And (amp; amp;), Or ()	79
13.5 Logical Not (!)	81
14 Numbers	83
14.1 JavaScript only has floating point numbers	83
14.2 Number literals	83
14.3 Number operators	84
14.4 Converting to number	86
14.5 Error values	87
14.6 Error value: NaN	87
14.7 Error value: Infinity	88
14.8 The precision of numbers: careful with decimal fractions	89
14.9 (Advanced)	89
14.10 Background: floating point precision	89
14.11 Integers in JavaScript	90
14.12 Bitwise operators	92
14.13 Quick reference: numbers	94
15 Math	101
15.1 Data properties	101
15.2 Exponents, roots, logarithms	102
15.3 Rounding	103
15.4 Trigonometric Functions	104
15.5 asm.js helpers	106
15.6 Various other functions	106
15.7 Sources	107
16 Strings	109
16.1 Plain string literals	109
16.2 Accessing characters and code points	110
16.3 String concatenation via +	110
16.4 Converting to string	111
16.5 Comparing strings	113
16.6 JavaScript characters vs. Unicode code points	113
16.7 Quick reference: Strings	115
17 Using template literals and tagged templates	123
17.1 Disambiguation: "template"	123
17.2 Template literals	124
17.3 Tagged templates	124
17.4 Raw string literals	126
17.5 (Advanced)	126
17.6 Multi-line template literals and indentation	127
17.7 Simple templating via template literals	128
17.8 Further reading	130

[Click here to download full PDF material](#)