
Table of Contents

Preface	xi
1. Introduction	1
Patterns	1
JavaScript: Concepts	3
Object-Oriented	3
No Classes	4
Prototypes	4
Environment	5
ECMAScript 5	5
JSLint	6
The Console	6
2. Essentials	9
Writing Maintainable Code	9
Minimizing Globals	10
The Problem with Globals	11
Side Effects When Forgetting var	12
Access to the Global Object	13
Single var Pattern	13
Hoisting: A Problem with Scattered vars	14
for Loops	15
for-in Loops	17
(Not) Augmenting Built-in Prototypes	19
switch Pattern	20
Avoiding Implied Typecasting	21
Avoiding eval()	21
Number Conversions with parseInt()	23
Coding Conventions	23
Indentation	24
Curly Braces	24

Opening Brace Location	25
White Space	26
Naming Conventions	28
Capitalizing Constructors	28
Separating Words	28
Other Naming Patterns	29
Writing Comments	30
Writing API Docs	30
YUIDoc Example	31
Writing to Be Read	34
Peer Reviews	35
Minify...In Production	36
Run JSLint	37
Summary	37
3. Literals and Constructors	39
Object Literal	39
The Object Literal Syntax	40
Objects from a Constructor	41
Object Constructor Catch	41
Custom Constructor Functions	42
Constructor's Return Values	43
Patterns for Enforcing new	44
Naming Convention	45
Using that	45
Self-Invoking Constructor	46
Array Literal	46
Array Literal Syntax	47
Array Constructor Curiousness	47
Check for Array-ness	48
JSON	49
Working with JSON	49
Regular Expression Literal	50
Regular Expression Literal Syntax	51
Primitive Wrappers	52
Error Objects	53
Summary	54
4. Functions	57
Background	57
Disambiguation of Terminology	58
Declarations Versus Expressions: Names and Hoisting	59
Function's name Property	60

Function Hoisting	61
Callback Pattern	62
A Callback Example	63
Callbacks and Scope	64
Asynchronous Event Listeners	66
Timeouts	66
Callbacks in Libraries	67
Returning Functions	67
Self-Defining Functions	68
Immediate Functions	69
Parameters of an Immediate Function	70
Returned Values from Immediate Functions	71
Benefits and Usage	72
Immediate Object Initialization	73
Init-Time Branching	74
Function Properties—A Memoization Pattern	76
Configuration Objects	77
Curry	79
Function Application	79
Partial Application	80
Currying	81
When to Use Currying	83
Summary	84
5. Object Creation Patterns	87
Namespace Pattern	87
General Purpose Namespace Function	89
Declaring Dependencies	90
Private Properties and Methods	92
Private Members	92
Privileged Methods	93
Privacy Failures	93
Object Literals and Privacy	94
Prototypes and Privacy	95
Revealing Private Functions As Public Methods	96
Module Pattern	97
Revealing Module Pattern	99
Modules That Create Constructors	100
Importing Globals into a Module	101
Sandbox Pattern	101
A Global Constructor	101
Adding Modules	103
Implementing the Constructor	104

Static Members	105
Public Static Members	105
Private Static Members	107
Object Constants	109
Chaining Pattern	110
Pros and Cons of the Chaining Pattern	111
method() Method	112
Summary	113
6. Code Reuse Patterns	115
Classical Versus Modern Inheritance Patterns	115
Expected Outcome When Using Classical Inheritance	116
Classical Pattern #1—The Default Pattern	117
Following the Prototype Chain	117
Drawbacks When Using Pattern #1	119
Classical Pattern #2—Rent-a-Constructor	120
The Prototype Chain	121
Multiple Inheritance by Borrowing Constructors	122
Pros and Cons of the Borrowing Constructor Pattern	123
Classical Pattern #3—Rent and Set Prototype	123
Classical Pattern #4—Share the Prototype	124
Classical Pattern #5—A Temporary Constructor	125
Storing the Superclass	126
Resetting the Constructor Pointer	127
Klass	128
Prototypal Inheritance	130
Discussion	132
Addition to ECMAScript 5	132
Inheritance by Copying Properties	133
Mix-ins	135
Borrowing Methods	136
Example: Borrow from Array	137
Borrow and Bind	137
Function.prototype.bind()	138
Summary	139
7. Design Patterns	141
Singleton	141
Using new	142
Instance in a Static Property	143
Instance in a Closure	144
Factory	146
Built-in Object Factory	148

Iterator	149
Decorator	151
Usage	151
Implementation	151
Implementation Using a List	154
Strategy	155
Data Validation Example	156
Façade	158
Proxy	159
An Example	160
Proxy As a Cache	167
Mediator	167
Mediator Example	168
Observer	171
Example #1: Magazine Subscriptions	171
Example #2: The Keypress Game	175
Summary	178
8. DOM and Browser Patterns	181
Separation of Concerns	181
DOM Scripting	183
DOM Access	183
DOM Manipulation	184
Events	185
Event Handling	186
Event Delegation	188
Long-Running Scripts	189
setTimeout()	189
Web Workers	190
Remote Scripting	190
XMLHttpRequest	191
JSONP	192
Frames and Image Beacons	195
Deploying JavaScript	196
Combining Scripts	196
Minifying and Compressing	197
Expires Header	197
Using a CDN	197
Loading Strategies	198
The Place of the <script> Element	199
HTTP Chunking	200
Dynamic <script> Element for Nonblocking Downloads	201
Lazy-Loading	203

Loading on Demand	203
Preloading JavaScript	205
Summary	206

Index	209
--------------------	------------