
Data Structures and Algorithms with JavaScript

Michael McMillan

Data Structures and Algorithms with JavaScript

by Michael McMillan

Copyright © 2014 Michael McMillan. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: 800-998-9938 or corporate@oreilly.com.

Editors: Brian MacDonald and Meghan Blanchette

Production Editor: Melanie Yarbrough

Copyeditor: Becca Freed

Proofreader: Amanda Kersey

Indexer: Ellen Troutman-Zaig

Cover Designer: Karen Montgomery

Interior Designer: David Futato

Illustrators: Rebecca Demarest and Cynthia Clarke Fehrenbach

March 2014: First Edition

Revision History for the First Edition:

2014-03-06: First release

See <http://oreilly.com/catalog/errata.csp?isbn=9781449364939> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *Data Structures and Algorithms with JavaScript*, the image of an amur hedgehog, and related trade dress are trademarks of O'Reilly Media, Inc.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc. was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

ISBN: 978-1-449-36493-9

[LSI]

Table of Contents

Preface.....	ix
1. The JavaScript Programming Environment and Model.....	1
The JavaScript Environment	1
JavaScript Programming Practices	2
Declaring and Initializing Variables	3
Arithmetic and Math Library Functions in JavaScript	3
Decision Constructs	4
Repetition Constructs	6
Functions	7
Variable Scope	8
Recursion	10
Objects and Object-Oriented Programming	10
Summary	12
2. Arrays.....	13
JavaScript Arrays Defined	13
Using Arrays	13
Creating Arrays	14
Accessing and Writing Array Elements	15
Creating Arrays from Strings	15
Aggregate Array Operations	16
Accessor Functions	17
Searching for a Value	17
String Representations of Arrays	18
Creating New Arrays from Existing Arrays	18
Mutator Functions	19
Adding Elements to an Array	19
Removing Elements from an Array	20

Adding and Removing Elements from the Middle of an Array	21
Putting Array Elements in Order	22
Iterator Functions	23
Non-Array-Generating Iterator Functions	23
Iterator Functions That Return a New Array	25
Two-Dimensional and Multidimensional Arrays	27
Creating Two-Dimensional Arrays	27
Processing Two-Dimensional Array Elements	28
Jagged Arrays	30
Arrays of Objects	30
Arrays in Objects	31
Exercises	33
3. Lists.....	35
A List ADT	35
A List Class Implementation	36
Append: Adding an Element to a List	37
Remove: Removing an Element from a List	37
Find: Finding an Element in a List	38
Length: Determining the Number of Elements in a List	38
toString: Retrieving a List's Elements	38
Insert: Inserting an Element into a List	39
Clear: Removing All Elements from a List	39
Contains: Determining if a Given Value Is in a List	40
Traversing a List	40
Iterating Through a List	41
A List-Based Application	42
Reading Text Files	42
Using Lists to Manage a Kiosk	43
Exercises	47
4. Stacks.....	49
Stack Operations	49
A Stack Implementation	50
Using the Stack Class	53
Multiple Base Conversions	53
Palindromes	54
Demonstrating Recursion	56
Exercises	57
5. Queues.....	59
Queue Operations	59

An Array-Based Queue Class Implementation	60
Using the Queue Class: Assigning Partners at a Square Dance	63
Sorting Data with Queues	67
Priority Queues	70
Exercises	72
6. Linked Lists.....	73
Shortcomings of Arrays	73
Linked Lists Defined	74
An Object-Based Linked List Design	75
The Node Class	75
The Linked List Class	76
Inserting New Nodes	76
Removing Nodes from a Linked List	78
Doubly Linked Lists	81
Circularly Linked Lists	85
Other Linked List Functions	86
Exercises	86
7. Dictionaries.....	89
The Dictionary Class	89
Auxiliary Functions for the Dictionary Class	91
Adding Sorting to the Dictionary Class	93
Exercises	94
8. Hashing.....	97
An Overview of Hashing	97
A Hash Table Class	98
Choosing a Hash Function	98
A Better Hash Function	101
Hashing Integer Keys	103
Storing and Retrieving Data in a Hash Table	106
Handling Collisions	107
Separate Chaining	107
Linear Probing	109
Exercises	111
9. Sets.....	113
Fundamental Set Definitions, Operations, and Properties	113
Set Definitions	113
Set Operations	114
The Set Class Implementation	114

More Set Operations	116
Exercises	120
10. Binary Trees and Binary Search Trees.....	121
Trees Defined	121
Binary Trees and Binary Search Trees	123
Building a Binary Search Tree Implementation	124
Traversing a Binary Search Tree	126
BST Searches	129
Searching for the Minimum and Maximum Value	130
Searching for a Specific Value	131
Removing Nodes from a BST	132
Counting Occurrences	134
Exercises	137
11. Graphs and Graph Algorithms.....	139
Graph Definitions	139
Real-World Systems Modeled by Graphs	141
The Graph Class	141
Representing Vertices	141
Representing Edges	142
Building a Graph	143
Searching a Graph	145
Depth-First Search	145
Breadth-First Search	148
Finding the Shortest Path	149
Breadth-First Search Leads to Shortest Paths	149
Determining Paths	150
Topological Sorting	151
An Algorithm for Topological Sorting	152
Implementing the Topological Sorting Algorithm	152
Exercises	157
12. Sorting Algorithms.....	159
An Array Test Bed	159
Generating Random Data	161
Basic Sorting Algorithms	161
Bubble Sort	162
Selection Sort	165
Insertion Sort	167
Timing Comparisons of the Basic Sorting Algorithms	168
Advanced Sorting Algorithms	170

The Shellsort Algorithm	171
The Mergesort Algorithm	176
The Quicksort Algorithm	181
Exercises	186
13. Searching Algorithms.....	187
Sequential Search	187
Searching for Minimum and Maximum Values	190
Using Self-Organizing Data	193
Binary Search	196
Counting Occurrences	200
Searching Textual Data	202
Exercises	205
14. Advanced Algorithms.....	207
Dynamic Programming	207
A Dynamic Programming Example: Computing Fibonacci Numbers	208
Finding the Longest Common Substring	211
The Knapsack Problem: A Recursive Solution	214
The Knapsack Problem: A Dynamic Programming Solution	215
Greedy Algorithms	217
A First Greedy Algorithm Example: The Coin-Changing Problem	217
A Greedy Algorithm Solution to the Knapsack Problem	218
Exercises	220
Index.....	221