

| | |
|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. Record Nr. | UNINA9910961401603321 |
| Autore | Stefanov Stoyan |
| Titolo | Object-oriented JavaScript // Stoyan Stefanov, Kumar Chetan Sharma |
| Pubbl/distr/stampa | Birmingham, UK, : Packt Publishing, c2013 |
| ISBN | 9781680153590 1680153595 9781849693134 1849693137 |
| Edizione | [2nd ed.] |
| Descrizione fisica | 1 online resource (382 p.) |
| Altri autori (Persone) | SharmaKumar Chetan |
| Disciplina | 005.117 |
| Soggetti | Computer software - Reusability JavaScript (Computer program language) Object-oriented programming (Computer science) Software patterns |
| Lingua di pubblicazione | Inglese |
| Formato | Materiale a stampa |
| Livello bibliografico | Monografia |
| Note generali | Includes index. |
| Nota di contenuto | Intro -- Object-Oriented JavaScript Second Edition -- Table of Contents -- Object-Oriented JavaScript Second Edition -- Credits -- About the Authors -- About the Reviewer -- www.PacktPub.com -- Support files, eBooks, discount offers and more -- Why Subscribe? -- Free Access for Packt account holders -- Preface -- What this book covers -- What you need for this book -- Who this book is for -- Conventions -- Reader feedback -- Customer support -- Errata -- Piracy -- Questions -- 1. Object-oriented JavaScript -- A bit of history -- Browser wars and renaissance -- The present -- The future -- ECMAScript 5 -- Object-oriented programming -- Objects -- Classes -- Encapsulation -- Aggregation -- Inheritance -- Polymorphism -- OOP summary -- Setting up your training environment -- WebKit's Web Inspector -- JavaScriptCore on a Mac -- More consoles -- Summary -- 2. Primitive Data Types, Arrays, Loops, and Conditions -- Variables -- Variables are case sensitive -- Operators -- Primitive data types -- Finding out the value type - the typeof operator -- Numbers -- Octal and hexadecimal numbers -- Exponent literals -- Infinity -- NaN -- Strings -- String conversions -- Special strings -- Booleans -- Logical operators -- |

Operator precedence -- Lazy evaluation -- Comparison -- Undefined and null -- Primitive data types recap -- Arrays -- Adding/updating array elements -- Deleting elements -- Arrays of arrays -- Conditions and loops -- The if condition -- The else clause -- Code blocks -- Checking if a variable exists -- Alternative if syntax -- Switch -- Loops -- While loops -- Do-while loops -- For loops -- For-in loops -- Comments -- Summary -- Exercises -- 3. Functions -- What is a function? -- Calling a function -- Parameters -- Predefined functions -- parseInt() -- parseFloat() -- isNaN() -- isFinite() -- Encode/decode URIs -- eval().

A bonus - the alert() function -- Scope of variables -- Variable hoisting -- Functions are data -- Anonymous functions -- Callback functions -- Callback examples -- Immediate functions -- Inner (private) functions -- Functions that return functions -- Function, rewrite thyself! -- Closures -- Scope chain -- Breaking the chain with a closure -- Closure #1 -- Closure #2 -- A definition and closure #3 -- Closures in a loop -- Getter/setter -- Iterator -- Summary -- Exercises -- 4. Objects -- From arrays to objects -- Elements, properties, methods, and members -- Hashes and associative arrays -- Accessing an object's properties -- Calling an object's methods -- Altering properties/methods -- Using the this value -- Constructor functions -- The global object -- The constructor property -- The instanceof operator -- Functions that return objects -- Passing objects -- Comparing objects -- Objects in the WebKit console -- console.log -- Built-in objects -- Object -- Array -- A few array methods -- Function -- Properties of function objects -- Prototype -- Methods of function objects -- Call and apply -- The arguments object revisited -- Inferring object types -- Boolean -- Number -- String -- A few methods of string objects -- Math -- Date -- Methods to work with date objects -- Calculating birthdays -- RegExp -- Properties of RegExp objects -- Methods of RegExp objects -- String methods that accept regular expressions as arguments -- search() and match() -- replace() -- Replace callbacks -- split() -- Passing a string when a RegExp is expected -- Error objects -- Summary -- Exercises -- 5. Prototype -- The prototype property -- Adding methods and properties using the prototype -- Using the prototype's methods and properties -- Own properties versus prototype properties -- Overwriting a prototype's property with an own property -- Enumerating properties. isPrototypeOf() -- The secret __proto__ link -- Augmenting built-in objects -- Augmenting built-in objects - discussion -- Prototype gotchas -- Summary -- Exercises -- 6. Inheritance -- Prototype chaining -- Prototype chaining example -- Moving shared properties to the prototype -- Inheriting the prototype only -- A temporary constructor - new F() -- Uber - access to the parent from a child object -- Isolating the inheritance part into a function -- Copying properties -- Heads-up when copying by reference -- Objects inherit from objects -- Deep copy -- object() -- Using a mix of prototypal inheritance and copying properties -- Multiple inheritance -- Mixins -- Parasitic inheritance -- Borrowing a constructor -- Borrow a constructor and copy its prototype -- Summary -- Case study - drawing shapes -- Analysis -- Implementation -- Testing -- Exercises -- 7. The Browser Environment -- Including JavaScript in an HTML page -- BOM and DOM - an overview -- BOM -- The window object revisited -- window.navigator -- Your console is a cheat sheet -- window.location -- window.history -- window.frames -- window.screen -- window.open()/close() -- window.moveTo() and window.resizeTo() -- window.alert(), window.prompt(), and window.confirm() -- window.setTimeout() and window.setInterval() -- window.document -- DOM --

Core DOM and HTML DOM -- Accessing DOM nodes -- The document node -- documentElement -- Child nodes -- Attributes -- Accessing the content inside a tag -- DOM access shortcuts -- Siblings, body, first, and last child -- Walk the DOM -- Modifying DOM nodes -- Modifying styles -- Fun with forms -- Creating new nodes -- DOM-only method -- cloneNode() -- insertBefore() -- Removing nodes -- HTML-only DOM objects -- Primitive ways to access the document -- document.write() -- Cookies, title, referrer, domain -- Events -- Inline HTML attributes.

Element Properties -- DOM event listeners -- Capturing and bubbling -- Stop propagation -- Prevent default behavior -- Cross-browser event listeners -- Types of events -- XMLHttpRequest -- Sending the request -- Processing the response -- Creating XMLHttpRequest objects in IE prior to Version 7 -- A is for Asynchronous -- X is for XML -- An example -- Summary -- Exercises -- 8. Coding and Design Patterns -- Coding patterns -- Separating behavior -- Content -- Presentation -- Behavior -- Example of separating behavior -- Asynchronous JavaScript loading -- Namespaces -- An Object as a namespace -- Namespaced constructors -- A namespace() method -- Init-time branching -- Lazy definition -- Configuration object -- Private properties and methods -- Privileged methods -- Private functions as public methods -- Immediate functions -- Modules -- Chaining -- JSON -- Design patterns -- Singleton -- Singleton 2 -- Global variable -- Property of the Constructor -- In a private property -- Factory -- Decorator -- Decorating a Christmas tree -- Observer -- Summary -- A. Reserved Words -- Keywords -- Future reserved words -- Previously reserved words -- B. Built-in Functions -- C. Built-in Objects -- Object -- Members of the Object constructor -- The Object.prototype members -- ECMAScript 5 additions to Object -- Array -- The Array.prototype members -- ECMAScript 5 additions to Array -- Function -- The Function.prototype members -- ECMAScript 5 additions to a function -- Boolean -- Number -- Members of the Number constructor -- The Number.prototype members -- String -- Members of the String constructor -- The String.prototype members -- ECMAScript 5 additions to String -- Date -- Members of the Date constructor -- The Date.prototype members -- ECMAScript 5 additions to Date -- Math -- Members of the Math object -- RegExp -- The RegExp.prototype members.

Error objects -- The Error.prototype members -- JSON -- Members of the JSON object -- D. Regular Expressions -- Index.

Sommario/riassunto

If you've limited or no experience with JavaScript, this book will put you on the road to being an expert. A wonderfully compiled introduction to objects in JavaScript, it teaches through examples and practical play. Think in JavaScript Make object-oriented programming accessible and understandable to web developers Apply design patterns to solve JavaScript coding problems Learn coding patterns that unleash the unique power of the language Write better and more maintainable JavaScript code Type in and play around with examples that can be used in your own scripts In Detail JavaScript is the behavior, the third pillar in today's paradigm that looks at web pages as something that consists of clearly distinguishable parts: content (HTML), presentation (CSS) and behavior (JavaScript). Using JavaScript, you can create not only web pages but also desktop widgets, browser and application extensions, and other pieces of software. It's a pretty good deal: you learn one language and then code all kinds of different applications. While there's one chapter specifically dedicated to the web browser environment including DOM, Events and AJAX tutorials, the rest is applicable to the other environments Many web developers have tried

coding or adopting some bits of JavaScript, but it is time to "man up" and learn the language properly because it is the language of the browser and is, virtually, everywhere. This book starts from zero, not assuming any prior JavaScript programming knowledge and takes you through all the in-depth and exciting futures hidden behind the facade. Once listed in the "nice to have" sections of job postings, these days the knowledge of JavaScript is a deciding factor when it comes to hiring web developers. After reading this book you'll be prepared to ace your JavaScript job interview and even impress with some bits that the interviewer maybe didn't know. You should read this book if you want to be able to take your JavaScript skills to a new level of sophistication.
