

Extracted from:

CoffeeScript

Accelerated JavaScript Development

This PDF file contains pages extracted from *CoffeeScript*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit <http://www.pragprog.com>.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printer versions; the content is otherwise identical.

Copyright © 2010 The Pragmatic Programmers, LLC.

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The
Pragmatic
Programmers

CoffeeScript

*Accelerated
JavaScript
Development*



Trevor Burnham

Foreword by Jeremy Ashkenas
edited by Michael Swaine

CoffeeScript

Accelerated JavaScript Development

Trevor Burnham

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



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 The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at <http://pragprog.com>.

The team that produced this book includes:

Michael Swaine (editor)
Potomac Indexing, LLC (indexer)
Kim Wimpsett (copyeditor)
David Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Copyright © 2011 Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-934356-78-4
Printed on acid-free paper.
Book version: P1.0—July 2011

JavaScript is born free, but until recently, everywhere it was in chains.

JavaScript had never been a very pleasant language to work in: terribly slow, implemented with different quirks in different browsers, stuck fast in the amber of time since the late 1990s. Perhaps you used it in the past to implement a dropdown menu or a reorderable list, but you probably didn't enjoy the experience.

Fortunately for us, the JavaScript of today is enjoying a well-deserved renaissance. Thanks to the tireless efforts of browser implementers, it's now the fastest mainstream dynamic language; it's present everywhere, from servers to Photoshop, and it's the only possible language you can use to program all angles of the web.

CoffeeScript is a little language that aims to give you easy access to the good parts of JavaScript: the first-class functions, the hash-like objects, even the much-misunderstood prototype chain. If we do our job right, you'll end up writing one-third less code in order to generate much the same JavaScript you would have written in the first place.

CoffeeScript places a high value on the readability of code and the elimination of syntactic clutter. At the same time, there's a fairly one-to-one correspondence between CoffeeScript and JavaScript, which means that there should be no performance penalty—in fact, many JavaScript libraries end up running faster after being ported to CoffeeScript due to some of the optimizations the compiler can perform.

You're fortunate to have picked up this book, because Trevor has been an enthusiastic contributor to CoffeeScript since the early days. Few people know more about the ins and outs of the language or the history of the debate behind language features and omissions than he does. This book is a gentle introduction to CoffeeScript led by an expert guide.

I'm looking forward to hearing about all of the exciting projects that I'm sure will come out of it, and—who knows—perhaps you'll be inspired to create a little language of your very own.

Jeremy Ashkenas, creator of CoffeeScript

April 2011