shortcut

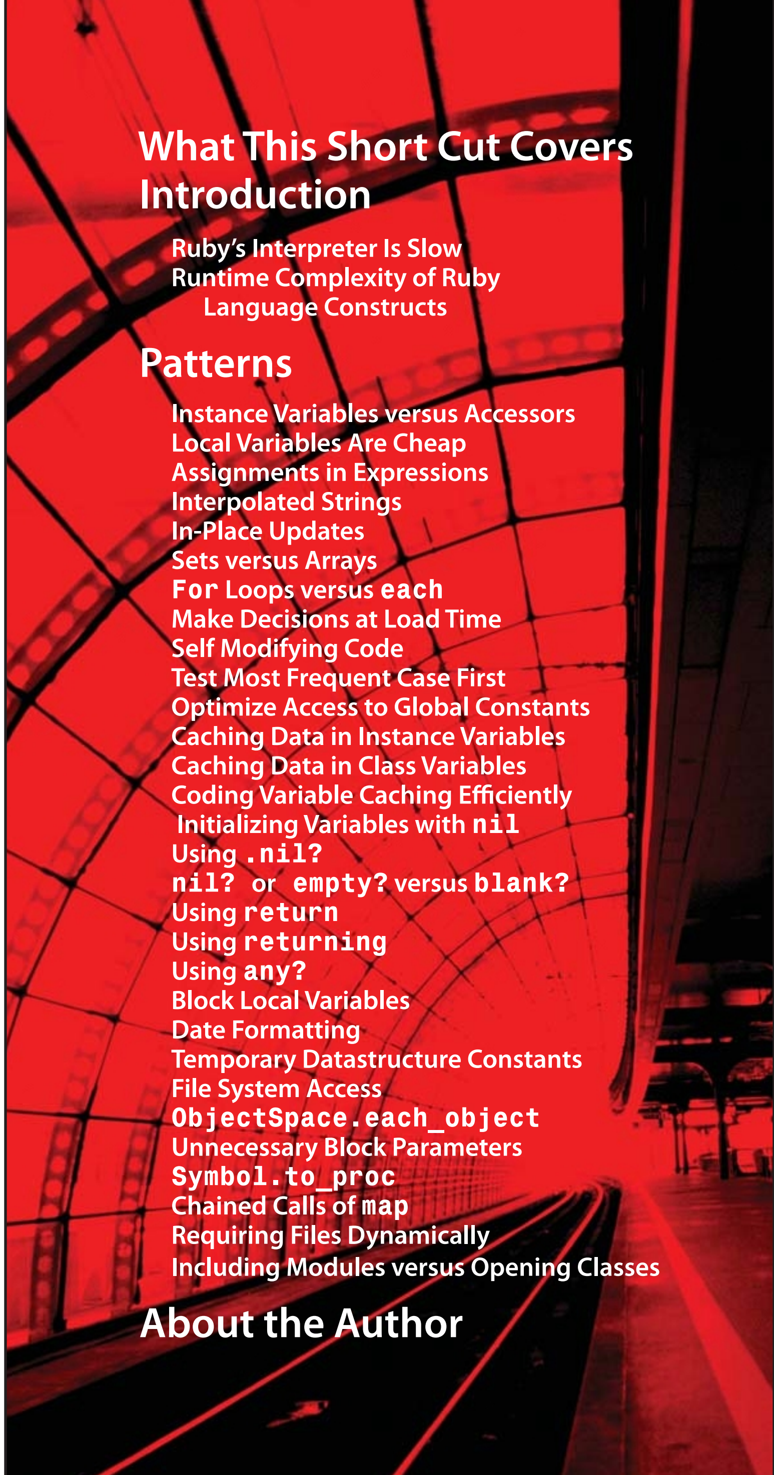
Addison-Wesley Professional Ruby Series

Writing Efficient Ruby Code

Dr. Stefan Kaes

Addison-Wesley
Pearson Education

www.informIT.com/Ruby



What This Short Cut Covers

Introduction

Ruby's Interpreter Is Slow
Runtime Complexity of Ruby Language Constructs

Patterns

Instance Variables versus Accessors
Local Variables Are Cheap
Assignments in Expressions
Interpolated Strings
In-Place Updates
Sets versus Arrays
For Loops versus each
Make Decisions at Load Time
Self Modifying Code
Test Most Frequent Case First
Optimize Access to Global Constants
Caching Data in Instance Variables
Caching Data in Class Variables
Coding Variable Caching Efficiently
Initializing Variables with nil
Using .nil?
nil? or empty? versus blank?
Using return
Using returning
Using any?
Block Local Variables
Date Formatting
Temporary Datastructure Constants
File System Access
ObjectSpace.each_object
Unnecessary Block Parameters
Symbol.to_proc
Chained Calls of map
Requiring Files Dynamically
Including Modules versus Opening Classes

About the Author

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this work, and the publisher was aware of a trademark claim, the designations have been printed with initial capital letters or in all capitals.

The author and publisher have taken care in the preparation of this work, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

Visit us on the Web: www.informIT.com

Copyright © 2008 Pearson Education, Inc.

This product is offered as an Adobe Reader™ PDF file and does not include digital rights management (DRM) software. While you can copy this material to your computer, you are not allowed to share this file with others.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc.
Rights and Contracts Department
501 Boylston Street, Suite 900
Boston, MA 02116
Fax: (617) 848-7047

ISBN-13: 978-0-321-54003-4
ISBN-10: 0-321-54003-4

Second release, February 2008

Writing Efficient Ruby Code (Digital Short Cut)

Table of Contents

What This Short Cut Covers

Introduction

- Rubys Interpreter Is Slow
- Runtime Complexity of Ruby Language Constructs

Patterns

- Instance Variables versus Accessors
- Local Variables Are Cheap
- Assignments in Expressions
- Interpolated Strings
- In-Place Updates
- Sets versus Arrays
- For Loops versus each
- Make Decisions at Load Time
- Self Modifying Code
- Test Most Frequent Case First
- Optimize Access to Global Constants
- Caching Data in Instance Variables

Pearson

Table of Contents

- Caching Data in Class Variables
- Coding Variable Caching Efficiently
- Initializing Variables with nil
- Using .nil?
- nil? or empty? versus blank?
- Using return
- Using returning
- Using any?
- Block Local Variables
- Date Formatting
- Temporary Datastructure Constants
- File System Access
- ObjectSpace.each_object
- Unnecessary Block Parameters
- Symbol.to_proc
- Chained Calls of map
- Requiring Files Dynamically
- Including Modules versus Opening Classes

About the Author

Pearson