

Functional Programming for the Web



ClojureScript

Up and Running

O'REILLY[®]

*Stuart Sierra &
Luke VanderHart*

ClojureScript: Up and Running

Learn how to build complete client-side applications with ClojureScript, the Clojure language variant that compiles to optimized JavaScript. This hands-on introduction shows you how ClojureScript not only has similarities to JavaScript—without the flaws—but also supports the full semantics of its parent language. You'll delve into ClojureScript's immutable data structures, lazy sequences, first-class functions, macros, and support for JavaScript libraries.

No previous experience with Clojure or ClojureScript is necessary. If you're familiar with JavaScript, HTML, CSS, and the DOM, you'll quickly discover that ClojureScript has the same reach as JavaScript, but with more power.

- Start writing ClojureScript code with the Leiningen build system
- Learn how the ClojureScript compiler works to produce optimized JavaScript
- Use JavaScript functions and libraries directly from ClojureScript code
- Explore functions in Clojure's sequence library, such as map, reduce, and filter
- Use macros to define new control structures or embed domain-specific languages
- Compile manually or script your own workflow with ClojureScript's compiler tools
- Integrate ClojureScript with Clojure on the JVM to build powerful client-server applications

Purchase the ebook edition of this O'Reilly title at oreilly.com and get free updates for the life of the edition. Our ebooks are optimized for several electronic formats, including PDF, EPUB, Mobi, and DAISY—all DRM-free.

Twitter: [@oreillymedia](https://twitter.com/oreillymedia)
facebook.com/oreilly

US \$19.99

CAN \$20.99

ISBN: 978-1-449-32743-9



O'REILLY[®]
oreilly.com

ClojureScript: Up and Running

Stuart Sierra and Luke VanderHart

O'REILLY®

Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

ClojureScript: Up and Running

by Stuart Sierra and Luke VanderHart

Copyright © 2013 Stuart Sierra, Luke VanderHart. 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.

Editor: Meghan Blanchette

Production Editor: Rachel Steely

Proofreader: Kara Ebrahim

Cover Designer: Karen Montgomery

Interior Designer: David Futato

Illustrator: Rebecca Demarest

Revision History for the First Edition:

2012-10-24 First release

See <http://oreilly.com/catalog/errata.csp?isbn=9781449327439> for release details.

Nutshell Handbook, the Nutshell Handbook logo, and the O'Reilly logo are registered trademarks of O'Reilly Media, Inc. *ClojureScript: Up and Running*, the image of a yellow-back duiker, 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: 978-1-449-32743-9

[LSI]

Table of Contents

Preface	vii
1. Introduction: Why ClojureScript?	1
The Rise of Browser Applications	1
The Rise of JavaScript	2
The Need for a Better Language	2
Introducing ClojureScript	3
2. Hello World	5
Leiningen	6
Installing Leiningen on OS X and Linux	6
Installing Leiningen on Windows	7
Using lein-cljsbuild	7
Getting Started with the REPL	8
Compiling a ClojureScript File to JavaScript	9
Running ClojureScript in the Browser	11
Other Capabilities of lein-cljsbuild	11
3. The Compilation Process	13
Architecture	13
Google Closure Compiler	13
The Google Closure Library	15
ClojureScript and Google Closure	16
The Compilation Pipeline	16
How to Compile	17
Compiling ClojureScript	17
Compilation in Depth	19
Compilation Sources	19
Compilation and Optimization Options	19
Other Compilation Options	23

Summary	23
4. ClojureScript Basics.....	25
ClojureScript versus Clojure	25
Expressions and Side Effects	26
Syntax and Data Structures	26
Symbols and Keywords	27
Data Structures	27
Special Forms and Definitions	28
Functions	29
Multi-Arity Functions	30
Variadic Functions	30
Local Bindings	30
Destructuring	31
Closures	31
Flow Control	32
Conditional Branching	32
JavaScript Interop	35
The js Namespace	35
Methods and Fields	36
Constructor Functions	36
Scope of this	37
Exceptions	38
Summary	38
5. Data and State.....	39
Primitives	39
Strings	40
Keywords	40
Symbols	40
Characters	41
Numbers	41
Booleans	41
Functions	41
nil	42
Data Structures	42
Collection Types	43
Immutability	46
Persistence	47
Identity and State	48

Atoms	48
6. Sequences.....	51
The Sequence Abstraction	51
Lazy Sequences	52
Letting Go of the Head	53
The Sequence API	54
map	54
reduce	54
filter	55
Other Useful Sequence Functions	55
7. Namespaces, Libraries, and Google Closure.....	57
Namespaces	57
Using Namespaces	58
Using Namespaces Effectively	59
The Implementation of Namespaces	60
Advanced Compilation Mode	61
Consuming Libraries	62
ClojureScript Libraries	62
JavaScript Libraries	63
Creating Libraries	66
For Consumption by ClojureScript	67
For Consumption by JavaScript	68
8. Macros.....	69
Code as Data	69
Writing Macros	69
Syntax-Quote	71
Auto-Gensyms	71
Using Macros	72
When to Write Macros	72
Summary	73
9. Development Process and Workflow.....	75
Installing ClojureScript	75
Checking Out from Source Control	76
Downloading a Compressed Archive	76
Installing Dependencies	77
The Built-In Tools	77
Command-Line Compilation	77
Clojure REPL	78

ClojureScript REPL	78
The Browser REPL	78
Setting Up the Browser REPL	79
Additional lein-cljsbuild Features	82
Launching a Browser REPL	82
Custom bREPL Launch Commands	83
Hooking Into Default Leiningen Tasks	83
Testing ClojureScript Code	84
Including ClojureScript in JAR Files	85
Compiling the Same Code as Clojure and ClojureScript	85
10. Integration with Clojure.....	87
AJAX	87
The Reader and Printer	88
Example Client-Server Application	89
Extending the Reader	93
User-Defined Tagged Literals	93
Sharing Code	94
Summary	95
A. Libraries.....	97

Who Should Read This Book

This book is for software developers who want to learn how to get started using ClojureScript to build web browser applications. This book will not assume any prior knowledge of ClojureScript. We do assume that you have at least a basic working knowledge of the core JavaScript language. For the sections of this book that deal with ClojureScript in a web browser, we assume you are familiar with HTML, CSS, the DOM, and how they are manipulated in JavaScript.

While this book will not assume any prior knowledge of Clojure, it is not designed to be a comprehensive reference to the Clojure programming language. We will explain Clojure language concepts in ClojureScript as they become important, but we also recommend picking up a book on Clojure for a more thorough guide to the language. The authors of this book wrote *Practical Clojure* (Apress, 2010) and O'Reilly has released *Clojure Programming* by our friends Chas Emerick, Brian Carper, and Christophe Grand.

How to Use This Book

This book is both a how-to guide for using ClojureScript and a tutorial on the language itself. We have arranged the chapters in what we felt was the best order for someone who is completely new to the language but wants to get started quickly. If you already know Clojure or ClojureScript and just want advice on development tools and workflow, focus on Chapters 2, 3, 7, 9, and 10. If you want to dive into the language right away, start with Chapters 4 through 6 before reading about the development process.

Chapter 1, Introduction: Why ClojureScript?

In this chapter, we lay out the motivation for ClojureScript: why it exists and what role it is designed to fill.

Chapter 2, Hello World

In this chapter, we work through a complete, albeit trivial, ClojureScript application. We introduce Leiningen, the lein-cljsbuild plug-in, and how to use ClojureScript in an HTML page. We save explanation for later chapters, but this chapter should be enough to get your first ClojureScript code “up and running.”

Chapter 3, The Compilation Process

This chapter goes into the ClojureScript compiler in detail, explaining how it works, most of the configuration options it supports, and how it integrates with the Google Closure Compiler.

Chapters 4 through 6 cover the basics of the ClojureScript language itself. Although not a complete guide to every corner of the language, they cover most of the features that are required for everyday programming. Because ClojureScript and Clojure are so similar, we recommend books about Clojure to learn more about the language.

Chapter 4, ClojureScript Basics

This chapter introduces the essential syntax and control structures of the ClojureScript language including functions, bindings, scope, and interoperability with JavaScript.

Chapter 5, Data and State

This chapter covers the primitive and composite data structures of ClojureScript, and shows how to work with them in programs. In particular, it explains ClojureScript’s approach to immutability and state management.

Chapter 6, Sequences

This chapter introduces Lazy Sequences, an important data structure in ClojureScript that makes up a substantial portion of the standard library.

Chapter 7, Namespaces, Libraries, and Google Closure

This chapter covers namespaces as a feature of the ClojureScript language and also explains how files are organized in ClojureScript projects. We go into detail about how the Google Closure Compiler affects the use of libraries in ClojureScript projects, and provide a detailed flowchart for determining how best to use any particular library.

Chapter 8, Macros

This chapter introduces macros, an advanced language feature provided by ClojureScript.

Chapter 9, Development Process and Workflow

This chapter covers a variety of alternative methods for working with ClojureScript code apart from the workflow we have used elsewhere in the book. We demonstrate some tools packaged with ClojureScript itself, including command-line compilation scripts and the ClojureScript Browser REPL (bREPL).

Chapter 10, Integration with Clojure

This chapter briefly demonstrates what can be achieved by combining Clojure and ClojureScript in the same application.

Conventions Used in This Book

The following typographical conventions are used in this book:

Italic

Indicates new terms, URLs, email addresses, filenames, and file extensions.

Constant width

Used for program listings, as well as within paragraphs to refer to program elements such as variable or function names, databases, data types, environment variables, statements, and keywords.

Constant width bold

Shows commands or other text that should be typed literally by the user.

Constant width italic

Shows text that should be replaced with user-supplied values or by values determined by context.



This icon signifies a tip, suggestion, or general note.



This icon indicates a warning or caution.

Using Code Examples

This book is here to help you get your job done. In general, you may use the code in this book in your programs and documentation. You do not need to contact us for permission unless you're reproducing a significant portion of the code. For example, writing a program that uses several chunks of code from this book does not require permission.

Selling or distributing a CD-ROM of examples from O’Reilly books does require permission. Answering a question by citing this book and quoting example code does not require permission. Incorporating a significant amount of example code from this book into your product’s documentation does require permission.

We appreciate, but do not require, attribution. An attribution usually includes the title, author, publisher, and ISBN. For example: “*ClojureScript: Up and Running* by Stuart Sierra and Luke VanderHart (O’Reilly). Copyright 2013 Stuart Sierra and Luke VanderHart, 978-1-449-32743-9.”

If you feel your use of code examples falls outside fair use or the permission given above, feel free to contact us at permissions@oreilly.com.

Safari® Books Online



Safari Books Online (www.safaribooksonline.com) is an on-demand digital library that delivers expert **content** in both book and video form from the world’s leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of **product mixes** and pricing programs for **organizations**, **government agencies**, and **individuals**. Subscribers have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O’Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and dozens **more**. For more information about Safari Books Online, please visit us **online**.

How to Contact Us

Please address comments and questions concerning this book to the publisher:

O’Reilly Media, Inc.
1005 Gravenstein Highway North
Sebastopol, CA 95472
800-998-9938 (in the United States or Canada)
707-829-0515 (international or local)
707-829-0104 (fax)