Learn by doing: less theory, more results

JavaScript Testing

Test and debug JavaScript the easy way

Beginner's Guide

Liang Yuxian Eugene

[PACKT] PUBLISHING

JavaScript Testing

Beginner's Guide

Test and debug JavaScript the easy way

Liang Yuxian Eugene

[PACKT] PUBLISHING

BIRMINGHAM - MUMBAI

JavaScript Testing

Beginner's Guide

Copyright © 2010 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author nor Packt Publishing and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: August 2010

Production Reference: 1130810

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.
ISBN: 978-1-849510-00-4

www.packtpub.com

Cover Image by Vinayak Chittar (vinayak.chittar@gmail.com)

Credits

Author

Liang Yuxian Eugene

Reviewers

Chetan Akarte

Kenneth Geisshirt

Stefano Provenzano

Aaron Saray

Mihai Vilcu

Acquisition Editor

Steven Wilding

Development Editor

Tarun Singh

Technical Editors

Paramanand N. Bhat

Pooja Pande

Copy Editors

Lakshmi Menon

Janki Mathuria

Editorial Team Leader

Akshara Aware

Project Team Leader

Priya Mukherji

Project Coordinator

Vincila Colaco

Indexer

Hemangini Bari

Proofreader

Dirk Manuel

Production Coordinator

Shantanu Zagade

Cover Work

Shantanu Zagade

About the Author

Liang Yuxian Eugene enjoys solving difficult problems creatively in the form of building web applications by using Python/Django and JavaScript/JQuery. He also enjoys doing research related to the areas of recommendation algorithms, link analysis, data visualization, data mining, information retrieval, business intelligence, and intelligent user interfaces. He is currently pursuing two degrees, Business Administration and Computer Science at National Cheng Chi University (NCCU) at Taipei, Taiwan. Eugene has recently started a personal blog at `http://www.liangeugene.com`.

I want to thank all of the great folks at Packt Publishing for giving me the opportunity to write this book. This book would not be possible without the help, advice and timely correspondence of Steven Wilding, Tarun Singh, Vincila Colaco and Priya Mukherji of Packt Publishing.

I want to thank Professor Johannes K. Chiang (Department of Management of Information Systems, NCCU) and Professor Li Tsai Yen (Department of Computer Science, NCCU) for their unwavering generosity in providing both personal and professional advice to me whenever I needed it.

I want to thank my family and friends for their continued support.

Last but not the least, I want to thank Charlene Hsiao for her kind understanding and tireless support for me.

About the Reviewers

Chetankumar D. Akarte has been working in PHP, JavaScript and .Net for the last five years. He has worked extensively on both small scale and large scale PHP and .Net ecommerce, social networking, Wordpress and Joomla based web projects. Over the years, Chetan has been actively involved in the "Xfunda Developers Community". He has regularly blogged on Microsoft .NET technology at `http://www.tipsntracks.com`.

Chetan completed a Bachelor of Engineering degree in Electronics from the Nagpur University, India in 2006. He likes contributing to newsgroups, and forums. He has also written some articles for Electronics For You, DeveloperIQ, and Flash & Flex Developer's magazines.

Chetan lives in Navi Mumbai, India. You can visit his websites at `http://www.xfunda.com` and `http://www.tipsntracks.com`, or get in touch with him at `chetan.akarte@gmail.com`.

I would like to thank my sister Poonam and brother-in-law Vinay for their consistent support and encouragement. I would also like to thank Packt Publishing for providing me with the opportunity to do something useful, and especially my Project Coordinator Vincila Colaco for all of the valuable support.

Kenneth Geisshirt is a chemist by education and a geek by nature. He has been programing for more than 25 years–the last six years as a subcontractor. In 1990 Kenneth first used free software, and in 1992 turned to Linux as a primary operating system (officially Linux user no. 573 at the Linux Counter). He has written books about Linux, PAM, and Javascript–and many articles on open source software for computer magazines. Moreover, Kenneth has been a technical reviewer of books on Linux network administration and the Vim editor.

Stefano Provenzano is an Italian senior consultant and professional software engineer. Stefano has worked on several projects in different fields of computer science—3D realtime engines for PC and Playstation games, visual simulation and virtual prototyping, web applications, and system integration. In 2006, Stefano started his own software development and consulting company, Shin Software. Currently, Stefano is developing CRM and INTRANET applications by using PHP and Javascript.

I want to thank my wife Irene and my little son Davide.

Aaron Saray found love when he was eight. It was in the shapely form of a Commodore 64. From then on, he continued to devote his time to various programing languages from BASIC to Pascal, PHP to Javascript, HTML to CSS. Aaron is both an author of a PHP Design Patterns book and a technical editor of other PHP and Javascript books. He has also worked as a professional in the Web Development field for almost a decade, and comes with a solid history to provide his vast experience to the review of this book. You can find more about his work at his technical blog by visiting `http://aaronsaray.com/blog`.

As each book project becomes complete, I learn more about my industry and myself. I want to specifically thank my best friend for consistently reminding me that life is always better with balance.

Mihai Vilcu has had exposure to top technologies in testing for both automated and manual testing. "Software testing excellence" is the motto that drives Mihai's career". This includes functional and non-functional testing. Mihai was also involved over several years in large scale testing projects.

Some of the applications covered by Mihai in his career include CRMs, ERPs, billing platforms, rating, collection and business process management applications.

As software platforms are used intensely in many industries, Mihai has performed testing in fields like telecom, banking, healthcare, software development, and others.

Feel free to contact Mihai for questions regarding testing on his email: `mvilcu@mvfirst.ro`, or directly on his website at `www.mvfirst.ro`.

Table of Contents

Preface 1

Chapter 1: What is JavaScript Testing? 7

Where does JavaScript fit into the web page? 8

HTML Content 8

Time for action – building a HTML document 9

Styling HTML elements using its attributes 11

Specifying id and class name for an HTML element 12

Cascading Style Sheets 12

Time for action – styling your HTML document using CSS 14

Referring to an HTML element by its id or class name and styling it 18

Differences between a class selector and an id selector 19

Other uses for class selectors and id selectors 20

Complete list of CSS attributes 20

JavaScript providing behavior to a web page 20

Time for action – giving behavior to your HTML document 20

JavaScript Syntax 24

JavaScript events 26

Finding elements in a document 26

Putting it all together 28

The difference between JavaScript and server-side languages 29

Why pages need to work without JavaScript 30

What is testing? 31

Why do you need to test? 31

Types of errors 32

Loading errors 33

Time for action – loading errors in action 33

Partially correct JavaScript 34

Time for action – loading errors in action 35

Runtime errors 36

Time for action – runtime errors in action 36

Logic errors 37

Time for action – logic errors in action **38**
Some advice for writing error-free JavaScript **40**
Always check for proper names of objects, variables, and functions 40
Check for proper syntax 40
Plan before you code 40
Check for correctness as you code 40
Preventing errors by choosing a suitable text editor 41
Summary **41**

Chapter 2: Ad Hoc Testing and Debugging in JavaScript **43**
The purpose of ad hoc testing–getting the script to run **44**
What happens when the browser encounters an error in JavaScript **44**
Browser differences and the need to test in multiple browsers **45**
Time for action – checking for features and sniffing browsers **46**
Testing browser differences via capability testing 47
Time for action – capability testing for different browsers **48**
Are you getting the correct output and putting values in the correct places? **50**
Accessing the values on a form 50
Time for action – accessing values from a form **51**
Another technique for accessing form values 54
Accessing other parts of the web page 55
Time for action – getting the correct values in the correct places **55**
Does the script give the expected result **65**
What to do if the script doesn't run? **65**
Visually inspecting the code 66
Using alert() to see what code is running 66
Using alert() to see what values are being used 67
Time for action – using alert to inspect your code **67**
A less obtrusive way to check what code is running and the values used 71
Time for action – unobtrusively checking what values are used **72**
Commenting out parts of the script to simplify testing **75**
Time for action – simplifying the checking process **76**
Timing differences–making sure that the HTML is there before interacting with it **77**
Why ad hoc testing is never enough **78**
Summary **79**

Chapter 3: Syntax Validation **81**
The difference between validating and testing **82**
Code that is valid but wrong–validation doesn't find all the errors 83
Code that is invalid but right 83
Code that is invalid and wrong–validation finds some errors that might be difficult to spot any other way 83

Code quality **83**
HTML and CSS needs to be valid before you start on JavaScript 84
What happens if you don't validate your code 85
Color-coding editors–how your editor can help you to spot validation errors 87
Common errors in JavaScript that will be picked up by validation **89**
JSLint–an online validator **90**
Time for action – using JSLint to spot validation errors **91**
Valid code constructs that produce validation warnings **92**
Should you fix valid code constructs that produce validation warnings? 92
What happens if you don't fix them 93
How to fix validation errors **93**
Error—missing "use strict" statement 94
Time for action – fixing "use strict" errors **94**
Error—unexpected use of ++ 94
Time for action – fixing the error of "Unexpected use of ++" **95**
Error—functions not defined 96
Time for action – fixing the error of "Functions not defined" **96**
Too many var statements 97
Time for action – fixing the error of using too many var statements **98**
Expecting <\/ instead of <\ 100
Time for action – fixing the expectation of '<\/' instead of '</' **101**
Expected '===' but found '==' 102
Time for action – changing == to === **102**
Alert is not defined 102
Time for action – fixing "Alert is not defined" **103**
Avoiding HTML event handlers 103
Time for action – avoiding HTML event handlers **104**
Summary of the corrections we have done 106
JavaScript Lint–a tool you can download **112**
Challenge yourself–fix the remaining errors spotted by JSLint 113
Summary **113**

Chapter 4: Planning to Test **115**

A very brief introduction to the software lifecycle **116**
The agile method 116
The agile method and the software cycle in action 117
Analysis and design 117
Implementation and testing 117
Deployment 117
Maintenance 117
Do you need a test plan to be able to test? **117**