

# Preface

You may wonder, why a new edition of *Perl by Example*? Perl 5 hasn't really changed that much; in fact, it's changed very little at all since the third edition of this book was published. And since Perl 6 hasn't been officially released, why not wait? Well, consider this. Let's say you bought a new Whirlpool washing machine six years ago. It's running perfectly. But since then, the mounds of laundry washed by that machine have come and gone. Now you're sporting a new trendy fashion, you have designer sheets and towels, and the detergent brand you use is hypoallergenic, nontoxic, and biodegradable, not available when you bought the washer. Even though Perl 5 has changed very little, the computer world has. It is always in a flux of new innovations, technologies, applications, and fads, and programs are being written to accommodate those changes. Whether analyzing data from the GenBank sequence database, writing applications for an iPhone, creating a personal blog on "myspace," or adjusting to the changes in a new Vista version of Windows, some computer program is involved, and very possibly it is a Perl program. Whatever the case, we like to keep up with the times. This new edition of *Perl by Example* was written for just that purpose.

As we speak, I am teaching Perl at the UCSC<sup>1</sup> extension in Sunnyvale, California, to a group of professionals coming from all around the Silicon Valley. I always ask at the beginning of a class, "So why do you want to learn Perl?" The responses vary from, "Our company has an auction site on the Web and I'm the webmaster. I need to use Perl and Apache to process our order information and send it to Oracle," or "I work in a genetics research group at Stanford and have to sift through and analyze masses of data, and I heard that if I learn Perl, I won't have to depend on programmers to do this," or "I'm a UNIX/Linux system administrator and our company has decided that all admin scripts should be converted to Perl," or "I just got laid off and heard that it's an absolute must to have Perl on my resume." And I am always amazed at the variety of people who show up: engineers, scientists, geneticists, meteorologists, managers, salespeople, programmers, techies, hardware guys, students, stockbrokers, administrators of all kinds,

---

1. University of California, Santa Cruz.

librarians, authors, bankers, artists—you name it. Perl does not exclude anyone. Perl is for everyone and it runs on everything.

No matter who you are, I think you'll agree that a picture is worth a thousand words, and so is a good example. *Perl by Example* is organized to teach you Perl from scratch with examples of complete, succinct programs. Each line of a script example is numbered, and important lines are highlighted in bold. The output of the program is then displayed with line numbers corresponding to the script line numbers. Following the output is a separate explanation for each of the numbered lines. The examples are small and to the point for the topic at hand. Since the backbone of this book was used as a student guide to a Perl course, the topics are modularized. Each chapter builds on the previous one with a minimum of forward referencing and a logical progression from one topic to the next. There are exercises at the end of the chapters. You will find all of the examples on the CD at the back of the book. They have been thoroughly tested on a number of major platforms.

*Perl by Example* is not just a beginner's guide but a complete guide to Perl. It covers many aspects of what Perl can do, from regular expression handling, to formatting reports, to interprocess communication. It will teach you about Perl and, in the process, a lot about UNIX and Windows. Since Perl was originally written on and for UNIX systems, some UNIX knowledge will greatly accelerate your learning curve, but it is not assumed that you are by any means a guru. Anyone reading, writing, or just maintaining Perl programs can greatly profit from this text.

Perl has a rich variety of functions for handling strings, arrays, the system interface, networking, and more. In order to understand how these functions work, background information concerning the hows, whys, and what-fors is provided before demonstrating functional sample programs. This eliminates continually wading through manual pages and other books to understand what is going on, what the arguments mean, and what the function actually does.

The appendices contain a complete list of functions and definitions, command-line switches, special variables, popular modules, and the Perl debugger; a bioinformatics tutorial to introduce *BioPerl*, and a tutorial covering *mod\_perl*, the fast way to create server side Perl scripts that replace the need for the Common Gateway Interface.

I have been teaching for the past thirty years and am committed to understanding how people learn. Having taught Perl now for more than 14 years, all over the world, I find that many new Perlers get frustrated when trying to teach themselves how to program. Most people seem to learn best from succinct little examples and practice. So I wrote a book to help myself learn and to help my students, and now to help you. As Perl has grown, so have my books. This latest, fourth, edition includes a new chapter on Perl and DBI with MySQL, a revised chapter on Perl objects, and new examples and explanations for the rest of the chapters to keep things current and interesting. The appendix material has been revised to include *BioPerl* and *mod\_perl*. In this book, you will not only learn Perl, but also save yourself a great deal of time. At least that's what my students and readers have told me. You be the judge.