
Preface... That No One Will Ever Read

Research indicates that virtually no one actually reads prefaces in computer books. In fact, this preface could be about the Loch Ness Monster for all most readers care.

But why don't people read prefaces? Is it because prefaces sometimes get too personal? (“...and I want to thank Sal, who made this book possible.”) Is it because the sentiments of nerdy authors can be a bit repulsive? (“...Sal, my SparcStation, was always there when I needed her.”) Is it because they are predictably self-deprecatory in an ironic way? (“...and I want to thank: [list of 250 noted experts goes here] for reviewing drafts of this manual. Naturally all mistakes are my own.”) No, when it comes right down to it, people avoid prefaces because they're boring.

It is also *de rigueur* for the nerdish author to explain, folksinger style, why he or she decided to write this piece. I guess this book sort of came to me while I was camping out in the glorious Rocky Mountains. I was gazing at pristine waterfalls and awe-inspiring sunsets when I said to myself, “This land must be preserved: I shall write a KornShell book.” Possessed by a demonic ecological fervor, I roared down that mountain in a snowmobile, fired up a 1200-watt workstation, and wrote it all down for you. I only ask that you use the power of the KornShell for good, rather than evil.

Through the years, I've been lucky enough to get lots of feedback from my readers. What do you think they ask for? More text? No. More syntax? Definitely not. When it comes right down to it, readers consistently ask for only one thing: *more examples*.

So, I listened to them and based this tutorial entirely on examples.

I feel that short, focused examples are more valuable than long, detailed examples, and I've written the book accordingly. The purpose of the book is not to make me look clever; it is to make you look clever. Nevertheless, I've sprinkled a few lengthy examples into the mix for those of you who enjoy that sort of thing.

When I was a mere lad, my family doctor pointed to a massive encyclopedia of pharmaceuticals and told me that he could treat 90 percent of all patients with only six different medicines. Similarly, although the KornShell is a very rich language, my goal is to focus on the common script ailments and to leave the beriberi treatment to specialists.

Beginners may be wondering, “Is this book too hard for me?” No, probably not. Programmers experienced in various languages, though not in the KornShell, are probably wondering, “Is this book too wimpy for me?” No, probably not. To help guide both kinds of readers through the book, I've marked certain sections as being more appropriate for one group or the other. Many beginners need an explanation of loops, while experienced programmers need to know how KornShell loops differ from loops in other languages.

Since the KornShell runs on all sorts of operating systems, I've tried to keep operating system dependencies to a bare minimum. Nevertheless, I do assume that readers are either working directly on the UNIX¹ operating system or indirectly on UNIX through a package like U/WIN.

My three primary reviewers were KornShell gurus David Korn, Steve Sommars, and Glenn Fowler, all of whom gave fantastic and diplomatic criticism.

I also received important criticism and help from Tom Barstow, Dave Beckedorff, Bart Hanlon, Warren Johnson, Ed Johnston, Mark Keil, Mike Kong, Alice Lynch, Dave Penfield, Will Roaf, Frank Rubinsky, John Weiss, Daryl Winters, Teri Witham, and other people whose names I have no doubt forgotten to list. Eric Eldred and Quentin Sullivan found so many bugs in early drafts that, and I mean this as a compliment, they should have a brand of insecticide named after them.

Judy Tarutz is my long-suffering editor. In person, Judy is a kind, gentle soul with an outrageous sense of humor. Give her a red pen, though, and she turns into Conan the Barbarian. Her savage red pen cut a bloody swath through early drafts of this book. If we should chance to meet up one day, dear reader, I'll tell you all the "weak" jokes she made me take out.

I'd also like to thank the members of the Academy.

A consortium of great minds—those minds owned and operated by Ted Ricks, John Wait, Jack Danahy, and Steve Spicer—led me to the KornShell and helped me define the material for this book.

Finally, I'd like to thank the person who really made this book possible: my wife and best friend, Marilyn.

Whoops, almost forgot. Naturally, all mistakes are my own.

Author's Note Regarding the ksh93 Edition

It has been a very long time since I wrote the ksh88 edition, entitled *KornShell Programming Tutorial*. I began to suspect as much when I noticed that of all the people acknowledged in this Preface, and I'm sure they were capital reviewers—intelligent, meticulous, helpful—the only familiar name was my wife's.

But how to resurrect a vanished book? All traces of the ksh88 edition had disappeared until 1997, when a fragment was found suspended in a piece of amber at a Chinese restaurant in Palo Alto. Working round the clock, scientists at Addison Wesley Longman were able to piece together most of the old book. When gaps were found, DNA sequences from Kernighan and Ritchie were substituted.

After reading *KornShell Programming Tutorial*, many readers asked me important questions such as, "How does a joke get to be in your book?" or "Couldn't you just write a preface next time and skip the rest of the stuff that came after it?" or "Did you hear the one about the traveling KornShell salesman and the system administrator's daughter?"

1. UNIX is a registered trademark of UNIX System laboratories, Inc.

Actually, creating humor for this book is not as haphazard as it may seem. All jokes are put through a rigorous five-step Beta testing procedure. I need to ensure that a joke is just as funny on the tenth reading as it is on the first. Although not all Beta testers installed the jokes, I did get enough feedback to toss out a ridiculous amount of material, for example, the original start of Chapter 1 went like this:

“I first saw David Korn writing shell scripts in a small club in the Village in 1946. The post-War euphoria was contagious and it wasn't long before the whole room was up and swaying to a pounding UNIX beat.”

In a second edition, the author is obliged to correct the occasional error or two that cropped up in the first edition. (A special note to all of you at the Strategic Air Command—you're absolutely right guys, it *was* supposed to be a comma, *not* a semicolon.) A pair of translators, Hideyuki Hayashi and Masataka Isoya, get the credit for finding the most mistakes in the ksh88 edition. These two were incredibly rigorous and could not have been more diplomatic. (“*Barry-San, every word on page 233 appears to be a typo.*”) Yep, the truth is out there, but a lot of it failed to work its way into the ksh88 edition. To improve accuracy, Sarah Tuttle wrote a validation suite that tests most of the sample KornShell scripts appearing in this book.

When I wrote *KornShell Programming Tutorial*, KornShell 88 was beginning to make KornShell 86 obsolete. As I go to press for *Hands On KornShell 93 Programming*, KornShell 93 is starting to overtake KornShell 88. KornShell 93 is far richer than KornShell 88, and I've focused this second edition completely on KornShell 93. If you liked KornShell 88, you're going to love KornShell 93.

Several reviewers strongly suggested a chapter on CGI programming with KornShell. As an avid CGI programmer, I happily agreed.

The CD accompanying this book contains the U/WIN package, created by David Korn and a cast of thousands, and now commercially marketed by Global Technologies Development. The package contains all the popular UNIX utilities—the `grep`'s and `awk`'s and `diff`'s you know and recognize—all ready to run on your Windows 95 machine.

Table P-1 on the next page shows how I've organized the book.

Table P-1. Chapter by Chapter

Chapter	Describes
1	How the KornShell stacks up against other programming languages.
2	Writing simple KornShell scripts, executing them, debugging them, and so on.
3	Data types like strings, integers, floats, structures, arrays, and such.
4	Mathematics, which is surprisingly rich for a shell language.
5	Pattern matching, including the five uniquely KornShell wildcards.
6	Conditional evaluation.
7	Looping. You'll want to read it over and over and over again.
8	Creating simple menus with the <code>select</code> statement.
9	Passing arguments on the command line.
10	Functions, including regular KornShell functions, autoloaded functions, and extension functions written in C.
11	Start-up files, environments, exporting variables, and the like.
12	Input and output.
13	Manipulating strings—an area in which the KornShell excels.
14	Reserved variables like <code>PATH</code> and <code>PS1</code> . There are dozens of these.
15	Sending and trapping signals; running scripts in the background.
16	Command line editing, repeating previous commands, and such.
17	CGI scripting tutorial. The KornShell is a terrific choice for your next CGI program.
App. A	A quick reference of all statements in the KornShell. This appendix will become more valuable to you as your KornShell experience increases.
App. B	A very short HTML tutorial; this is just enough HTML to help understand what's going on in Chapter 17.

Primary reviewers for this book include John Allen, Chris Baron, Dave Beckedorff, Ann Hall, Bobby Higgins, Ollie Jones, Jeff Korn, David Korn, Charlie Northrup, and Bob Weil. Thanks also to Jason Jones, Mike Hendrickson, Marina Lang, and Marilyn Rash at Addison Wesley Longman. There's a saying in Japan that goes something like, "Everybody should climb Mt. Fuji once, but only a fool would climb it a second time." Dave Beckedorff—thanks for reviewing my book a second time. Thanks to the wonderful staff at Open Market, Inc. for their collective Web expertise.

The success of *KornShell Programming Tutorial* was primarily due to the collective "tough love" of its reviewers. Thank you again. Special thanks to Judy Tarutz for an editorial job that still amazes me seven years later.

And to my wife Marilyn—your love, support, and amazing patience is what really made this book.