ter
Technology Electronic Reviews

Formerly *Telecommunications Electronic Reviews*

Technology Electronic Reviews (TER) is a publication of the Library and Information Technology Association.

Contents:

News from the *TER* Editorial Board:

by Sharon Rankin

Welcome to Frank Cervone, who joins *TER* as Publisher Relations Editor (Northwestern University).

The *TER Readership Survey* received a very small number of comments, so we have withdrawn the survey for this issue. It will reappear again in 2006 issues with new questions designed to collect your comments about improving this publication. LITA and the *TER* Editorial Board would like to hear from you about how to make this publication more relevant to your work and professional interests.

This year I will complete my term as TER Editor-in-Chief. LITA has announced its search for a new TER Editor. If you are interested in more information about the position and its responsibilities, the posting can be found at:
http://www.lita.org/ala/lita/newandnoteworthy/TERedad.htm
(http://www.lita.org/ala/lita/newandnoteworthy/TERedad.htm)

We are always looking for new reviewers. If you would like more information about becoming a reviewer, consult the *TER* web site or contact Florence Tang, Reviewer Relations Editor (tang_fy@mercer.edu). (mailto:tang_fy@mercer.edu)

REVIEW OF: Allen C. Benso. (2003). *Connecting Kids and The Web: A Handbook for Teaching Internet Use and Safety.* New York: Neal-Schuman.

by Michele Rothenberger

Allen Benson has been a teacher of web wisdom to K-12 teachers and students for more than ten years; this book is a distillation of his wisdom into a highly readable, accessible form.

The book is divided into five sections: Web Basics Parents, Teachers and Librarians Need to Know (chapters 1-3); Introducing Young Learners to the Web (chapters 4-7); Creating Web Fun for All Ages (chapters 8-11); Ready-to-Go Web Learning Activities for All Ages (chapters 12-14); and How to Make Teens Web Masters (chapters 15-18).

"Web Basics" is a broad topic, but Section 1 clearly and concisely covers the major points. Chapter 1 topics include the difference between the Internet and the web, domain names, a list (with URLs) of organizations that maintain some control over the Internet (W3C, ICANN, IAB, etc). Chapter 2 covers safety and security: rules for young internet users such as not disclosing personal information, passwords, virus protection and a brief overview of CIPA and COPPA. Chapter 3, "Staying Current," is particularly useful; rather than attempt a listing of specific new items, which would be rapidly outdated, the chapter wisely provides URLs for a wealth of "what's new" websites that help the curious user keep up to date.

Section 2, "Introducing Young Learners to the Web," is somewhat of a misnomer since Chapter 4 covers web and browser basics and will be useful to all ages. Chapter 5 is a collection of 22 of the author's personal favorites, some educational and some not, from the White House's official website to the home page for the Calvin and Hobbes cartoon. From a teacher's perspective, Chapter 6 is one of the best in the book: virtual tours and field trips are

a brilliant addition to a teacher's repertoire. Tramline (http://www.tramline.com (http://www.tramline.com)) is an especially stellar find: a list of links to 20+ virtual field trips (with grade levels), and "Tourmaker" software for teachers to create their own virtual field trips. Sites for online museums and exhibits as well as pen pals (both email and the pencil-and-paper variety) are also given. Chapter 7 is a reprise of Chapter 5 - a collection of just plain fun sites, divided K-6 and 7-12.

Section 3 (chapters 8-11) covers email and chat (Chapter 8) and mailing lists (Chapter 9), including an excellent resource for finding listservs geared for kids. Chapter 10 is an ambitious (perhaps too ambitious) attempt to cover all types of multimedia. Chapter 11's overview of HTML covers basic terminology and HTML elements, but beware that the examples don't conform to strict HTML or XML formatting (e.g., end tags are omitted), and instructions on validating HTML code are not given and would be useful (e.g. W3C's validator, http://validator.w3.org (http://validator.w3.org)).

Section 4 (chapters 12-14) explores a variety of topics and sites useful for research on the web. Chapter 12 does a sterling job covering subject directories and search engines, outlining good searching techniques, and explaining more sophisticated search skills such as Boolean searching, truncation, and proximity operators. Chapter 13 details library resources on the Internet, including both online access to local brick and mortar libraries and libraries that only exist online such as the Internet Public Library. Chapter 14 provides a number of online sources for news, magazines, journals, and books, and covers both alternative and mainstream media. Chapter 15 lists a number of excellent reference sites such as Encyclopedia Britannica and InfoMine; for each site the book explains what it's best used for and proposes appropriate search strategies.

Section 5, "How to Make Teens Web Masters," includes chapters 15-18, and is the most unusual part of the book. Most students won't encounter UNIX, telnet, or ftp outside of a computer course, but this section provides an introduction that may inspire some students to delve further. Chapter 16 discusses UNIX shell accounts, where to get them, and how to access email via UNIX. Chapter 17 explores telnet, bulletin board services (BBSs), and multi-user domains (MUDs, MOOs), including Pueblo, a long-established online "community" that has been the subject of a number of articles. Chapter 18 covers newsgroups, but curiously does not mention many of the other online group services such as YahooGroups or Delphi Forums. Chapter 19 explains ftp - browser, separate FTP apps, and UNIX shell methods.

Summary

The introduction states that the book is intended to be used as "a reference source… [and] a lesson planner but most importantly as a constant source of flint to spark children's imaginations and dare them to learn." (intro, p. xv). It succeeds well at the latter but only partially at the first two. The book is crammed with websites ranging from the silly (The Yuckiest Site on the Internet) to the studious (Librarian's Index to the Internet) and even a

quick look will make you itch to try them all out. An accompanying CD, "The Link Farm," organizes most of the links given in the book into subject directory format for an easy-to-use quick reference. Teachers, parents and librarians will all find something for children of any age, there is a good balance between entertainment and educational sites, and extensive site references in the more technical sections make it easy to follow up on individual topics as needed or desired. One caveat: several of the sites given in the book require account creation and/or registration before using, and a few of the sites are of dubious value; for example, one site recommended for web research seems primarily designed to sell research and term papers.

Only nine lesson plans are provided; most are for grades 7+ and most are actually step-by-step procedures rather than actual lesson plans. However, any number of the sites mentioned throughout the book might easily be turned into lesson plans.

As a reference work, the book has good points and bad. One drawback is a somewhat scattered organization that makes it difficult to find all related information in a single place; for example, Online Public Access Catalogs (OPACs) are not discussed in the chapter on libraries, but rather in the chapter on telnet. UNIX is covered in chapter 16 but common UNIX commands are listed in chapter 17. There are also some notable omissions; for example, the section on research omits any discussion of database searching (e.g. Lexis/Nexis or EbscoHost).

The more technical chapters are uneven in depth. For example, the book goes into great detail on the various protocols for email (IMAP, SMTP, POP) but doesn't cover identifying and avoiding common email scams and hoaxes. Chapter 10's lengthy discussion of digital video covers storyboarding, shooting, editing and distribution, yet doesn't cover something as simple as connecting a digital camera to download photos.

Finally, to be a good reference source, a book needs to provide multiple access points so that a searcher can quickly locate desired information. The index is the best way to do this, and in this area the book falls sadly short. At just four pages for a 350+ pages book, the index is skimpy to begin with, but it also omits dozens of major topics covered in the book: privacy policies, file types, favorites, plug-ins, bookmarks, google, subject directories, databases, and Tramline, to name but a few.

Minor flaws aside, the book is still a valuable resource. The author's obvious enthusiasm and sense of fun infuse every page, so that reading any single chapter opens up dozens of possibilities for educating and entertaining.

Michele Rothenberger works in the Special Collections Department of Syracuse University.

REVIEW OF: Steven Holzner. (2004). *Perl Core Language: Little Black Book*. Scottsdale, AZ: Paraglyph Press.

by John Wynstra

Perl has been around since 1986, and continues to be a popular programming language. It is used for CGI programming in web applications and for system administration reporting tasks on Unix/Linux servers among other things. It is particularly strong in its ability to manipulate textual data. Even though it has UNIX origins, it is available and quite popular on the Windows platform via the ActiveState port which can be downloaded from the ActiveState website http://www.activestate.com/ (http://www.activestate.com/).

"PERL Core Language" can easily be categorized as a programming language reference book; however, it also serves as an excellent introduction to the language for the beginner. That cannot be said of many programming language reference books since they are organized for the purpose of providing quick access to topics by experienced programmers and not for the purpose of giving a logical, progressive overview of the language. This book strikes such a good balance between reference and overview that it earns high marks in my opinion for its usability and broad content coverage.

The book's greatest strengths are its organizational features. Before even opening the book, the reader will observe the black finger index on the edge of the pages that provides quick access to the chapters. Thumbing through the pages reveals ½ x 1 ½ black finder tabs on the edge of each page displaying chapter titles in white letters. These tabs become more useful over time as the reader becomes familiar with the contents of the book and goes back to revisit a previously read section.

The book is subdivided into 4 parts as follows: Part I "Perl Syntax", Part II "Built-In Resources", Part III "Perl Programming", and Part IV "CGI Programming". Each part contains four to seven chapters based on well-defined topics. Each chapter is further subdivided into two sections titled "In Brief" and "Immediate Solutions." The "In Brief" sections are introductory and theoretical, whereas the "Immediate Solutions" sections are hands on and practical. Each chapter contains 15-30 how-to-do-it Immediate Solutions covering no more than 1-2 pages per topic. Each Immediate Solution is titled, labeled and set off by bold lines at the top and bottom. In addition, the title page of each chapter contains a nicely formatted table of contents that utilizes alternating gray and white highlights to make it easy to scan quickly for specific Immediate Solutions.

In Part I, Perl Syntax, the author systematically but quickly builds a foundation for programming with Perl, starting with where to obtain it, how to install it, and how to run a script from the command line using various switches. Each topic is handled concisely and practically. The author moves right along through variables, lists, hashes, arrays, loops, control statements, regular expressions, and subroutines; and before you know it you have a pretty good foundation for writing working Perl scripts.

Part II, Built-In Resources, focuses on leveraging variables and functions that are included with Perl. Perl includes a library of common functions to save programmers the time of rewriting already existing code. In addition, it includes a number of variables that are cryptic, but important to know about for writing or understanding even simple scripts. This section defines these variables and functions, and shows how each is used in code.

Part III, Perl Programming, discusses topics that are a little more advanced. Among them are: data structures, packages and modules, classes and objects (object oriented programming), and Perl debugging. In the data structures chapter, the author moves the reader beyond simple one-dimensional arrays and hashes into multidimensional arrays and hashes, including arrays of arrays, arrays of hashes, hashes of hashes, and hashes of arrays. Packages, modules, classes, and objects are structures for organizing programs into well-defined and manageable code units. The two chapters about them are good introductions, but an advanced programmer will need more information to understand them fully. The Perl debugging chapter is very useful and thorough.

Part IV, CGI Programming, is all about writing dynamic web scripts utilizing the CGI.PM module. The first chapter in this section explains and demonstrates techniques for using this object-oriented module. It discusses starting and ending HTML documents as well as creating basic HTML components including form elements. The next chapter in this section diverts its attention to XML. XML is often associated with web programming, and so it seems logical that it be in this section, but the chapter's primary code example and the author's approach to the topic doesn't deal with web programming at all. The final two chapters of this section and book are collections of simple CGI programs, including a web counter, a guestbook, an email script, a web chat script, a cookie script, and a game (hangman). As part of his discussion of these examples, the author includes a short but very useful discussion on security issues associated with CGI programming.

As with most programming language books, this book contains a large number of code samples. The examples in the book were written and tested using Perl version 5.8.4. Occasionally the author points out information about version 6 (currently under development) if it is relevant. In just about every example, the author utilizes a handy technique of highlighting a few lines of code in grey to draw the reader's attention to specific functions, techniques or variables that were just discussed. The examples are illustrative for the most part and not taken from actual practical usage of the language. Most examples are very short, consisting of five to ten lines of code. Some examples extend over a few pages, especially near the end of

the book. It would have been handy to have these examples available on a book website, but practically speaking, the long code examples are so few that it would not have justified including a CDROM with the book.

I would highly recommend this book. It is well organized, concise, and well-balanced. It is not what I would consider an exhaustive reference book, but it is a handy collection of very usable tips, tricks, and core language information. I am particularly impressed with its many well thought out organizational features, and with its content coverage aimed at both beginners and experienced programmers.

John Wynstra is the Library Information Systems Specialist at the University of Northern Iowa.

REVIEW OF: Christopher Schmitt. (2004). *CSS Cookbook*. Sebastopol, CA: O'Reilly.

by Margaret Kelly

There are two kinds of cookbooks: the kind that teaches you how to cook, and the kind that assumes basic technical knowledge on your part and serves instead to stimulate the imagination. *CSS Cookbook* by Christopher Schmitt is of the second variety. It is one of the O'Reilly Cookbook series, which includes more than forty titles in a range of information technology topics. In its 250 pages *CSS Cookbook* provides a supply of handy ideas that can be implemented on websites through the use of Cascading Style Sheets (CSS). Though the preface promises a book that will "solve common problems associated with CSS-enabled web page designs," it scores as highly on "browse-ability" as on specific problem solving.

According to the end matter in the book, Christopher Schmitt has formal training in graphic design and has written works on CSS, HTML, and XHTML. A web search retrieves several articles published by him in the areas of web design and user experience. Though one interview calls him "a CSS guru," he appears to be better described as its tireless advocate [1].

CSS is a set of specifications that website authors can use to control how web browsers render text, images, and other media. The entire specification document can be found online [2]. It differs from HTML in that the stylistic features of the presentation of a page (colors, fonts, sizes) are separated from the structural markup (paragraphs, line breaks). These stylistic features are placed in the header of that page or in a separate file referenced within the

header where they can be activated by tags in the HTML. This simplifies the maintenance of large sites and promotes consistency across numerous pages. Several pages can reference the same stylesheet and several stylesheets can be used to control one page. No special software is required in order to begin coding: one only needs a web browser that supports CSS.

The book begins by covering basic text styles and proceeds through topics that include forms, tables, overall page design and browser workarounds. It is assumed that the user has a working knowledge of HTML and can publish a web page. Although the chapters appear to be arranged in order of increasing complexity, it is possible to locate and use a solution without reading all the preceding chapters.

The format of *CSS Cookbook* is simple and consistent throughout. For each application of CSS there is a problem statement, a screenshot of the desired effect, the code required to produce the effect, and a discussion of why and how the code works. This conforms to the style established by O'Reilly in the Cookbook series. The screenshots are especially helpful because they make it possible to browse for a problem solution for which one does not know the technical term.

Code examples are distinguished from the explanatory text by use of different fonts. Screenshots are reduced to about one quarter of their size on a 17" monitor, yet they are large enough to read. However, some of the examples lose their contrast in the black and white format, and no color images are used in the book. Discussions of color conclude with references to websites devoted to color use. Though this raises the question of potential dead links, most designers probably already have favorite online color charts or print references.

I tried several of the "recipes" and found that each one worked perfectly the first time. While I might have been a little lost had I previously known nothing about CSS, I am nevertheless a relative beginner in its use and I was able to learn a great deal from copying out the CSS blocks into my test site. An example is item 3.6, "Creating Nongraphical Menus with Rollovers." Rollovers are not my favorite web effect and yet, after using the easy directions in the book, I began to think of sensible and creative ways to use them.

Another example I used was item 6.6, "Sample design; an elegant calendar." This exercise was so long and complex that it was broken into steps. Through the process of copying the code for each step into a test file and viewing it in my browser I was able to see the elements of the example build upon one another. Once all the code was in place it was easy to adapt the colors and other style attributes to suit my own preferences.

If I had glanced quickly through *CSS Cookbook* at a bookstore I might have passed it up in favor of one that offered more colorful examples. That would have been a mistake. It is by actually trying the solutions that the reader comes to appreciate the value of this resource.

Though it is not a guide for the complete beginner, anyone who has produced a simple CSS-enabled web page can pick up this work and begin following the examples. Intermediate and advanced users also will discover new ways to approach problems.

References

[1] " *Design Industry Interviews: Top 10 Reasons to Learn CSS*." Retrieved April 5, 2005, from http://www.sessions.edu/newsletter/Schmitt_C/interview.html/ (http://www.sessions.edu/newsletter/Schmitt_C/interview.html) ."

[2] " *Cascading Style Sheets, level 2: CSS2 Specification."* Retrieved April 5, 2005, from http://www.w3.org/TR/REC-CSS2/ (http://www.w3.org/TR/REC-CSS2/) ."

Margaret Kelly is Systems Librarian at the University of Michigan in Ann Arbor.

REVIEW OF: Andrew M. St. Laurent. *Understanding open source & free software licensing*. (2004). Sebastopol, CA: O'Reilly. (Guide to navigating licensing issues in existing & new software).

by Brad Lee Eden

This book is designed to assist those involved in research and commercial projects that use open source and free software products, and in the understanding of the licenses that need to be signed and agreed to for the use of those products. In the preface, the author specifically mentions the projects and licenses that are covered in the book. These include: MIT, BSD, Apache versions 1.0 and 2.0, Academic Free License (AFL), GNU General Public License (GPL), GNU Lesser General Public License (LGPL), Mozilla Public License (MPL), Qt License, Artistic License, Creative Commons License, Sun Community Source License and Commercial Use Supplement, and the Microsoft Shared Source Initiative. Each of these licenses is examined clause by clause, including the original license text and explanation. Other issues examined in the book include cross-licensing, formation of a contract, and enforceability of warranty and other disclaimers. The author indicates that this book is written for programmers writing personal code libraries, managers who incorporate open source code in their businesses, and even lawyers evaluating open source and proprietary code.

The book contains seven chapters, which are self-contained such that one does not need to read the entire book in order to extrapolate information regarding specific licenses, if one chooses. Chapter 1 deals with open source licensing, contract, and copyright law; the foundations of each and a short section on warranties and patents. Chapter 2 looks closely at the MIT, BSD, Apache, and Academic Free licenses. Chapter 3 contains information on the GPL, LGPL, and Mozilla licenses. Chapter 4 focuses on Qt, Artistic, and Creative Commons licenses. Non-open sources, or proprietary licenses, are dealt with in Chapter 5, where the Microsoft and Sun licenses are examined. Chapter 6 discusses the legal obligations of using or publishing open source or free software, consequences of such use, and questions regarding the mixing of licenses. Chapter 7 examines the concept of licensing itself, how open source and free licenses have been used and are being used on software projects, and how to draft your own license. An appendix contains the Creative Commons Attribution-NoDerivs 2.0 license, which sets the terms under which this book may be freely distributed.

While not highly technical, this book does delve deeply into contract and license language and terminology, and would probably not be understandable to anyone but those involved in this highly semantic area. People who are involved in software projects that use open source or free software would find this book an essential resource and guide. It is well written, provides concise and simple explanations regarding the contracts indicated above, and is the only book that I know of that examines each of these contracts and licenses in such detail.

Bradford Lee Eden, Ph.D., is Head, Web and Digitization Services, University of Nevada, Las Vegas Libraries.

About TER

- Florence Tang, Mercer University, Atlanta (tang_fy@mercer.edu (mailto:tang_fy@mercer.edu))
- Laura Wrubel, University of Maryland (lwrubel@umd.edu (mailto:lwrubel@umd.edu))
- Michael Yukin, University of Nevada, Las Vegas (michael.yunkin@ccmail.nevada.edu (mailto:michael.yunkin@ccmail.nevada.edu))