

# ALLGEMEINES

Herbert Breunung

## Rezension - Perl komplett

Johan Vromans  
Perl Pocket Reference  
5. Auflage Juli 2011  
102 Seiten, englisch  
O'Reilly  
ISBN 978-1-449-30370-9 \$12,99  
E-Book: ISBN 978-1-4493-0813-1 \$7,99

Jürgen Plate  
Der Perl-Programmierer  
Perl lernen – Professionell anwenden – Lösungen nutzen  
1. Auflage Juli 2010  
1.232 S., Flexcover  
Hanser Verlag  
ISBN 978-3-446-41688-8  
€59,90

### Übersicht

Die Frühjahrsausgabe dieser gemütlichen Bücherecke geht der Frage nach, was ein vollständiges Perlbuch ausmacht. Doch zuvor ein kleiner Spaziergang auf den Markt. Die neuen Webstandards sind natürlich hier ein größeres Thema. O'Reilly [1] hat seine CSS-Taschenreferenz aktualisiert und das 600-seitige Buch *Head First HTML5 Programming* herausgegeben, vorerst nur auf englisch.

Galileo behandelt beides mit der neunstündigen Lehr-DVD mit dem schlichten Namen *HTML5 und CSS3* von Peter Kröner, hat aber auch "Fortgeschrittene CSS-Techniken" von Chao und Rudel mit Beispiele- und Video-DVD. Zum verwandten Web-Thema `NoSQL` brachte Hanser [2] ein gleichnamiges deutsches Buch auf den Weg und O'Reilly eines über Hadoops `NoSQL`-Datenbank *HBase* (englisch).

Wer auf der Ebene darunter unterwegs ist, muss Ahnung von den Transportprotokollen haben. Der mitp-Verlag brachte mit der *TCP/IP - Studienausgabe* ein detailreiches und mit 24,95€ recht günstiges Werk heraus. Im gleichen Haus erschien die Neuauflage des einer Taschenreferenz ähnlichen Buches *vim 7.3 GE-PACKT*.

Linux ist immer ein Thema. Das wusste auch der Verlag Addison-Wesley, der *Linux-Firewalls* von Ralf Spenneberg und *Linux 2012* drucken lässt. Galileo stellt dazu *Linux: Das umfassende Handbuch* und einen Schmöker (1024 Seiten+DVD) über *Apache 2.4* von Sascha Kersken vor. Zu guter Letzt hat O'Reilly nicht nur – wie letztens erwähnt – die aktualisierte *Einführung in Perl* übersetzt, sondern auch eine Initiative in Sachen Design und Softwarequalität gestartet.

Von den drei Büchern *Designed for Use* (Wie Sorge ich dafür, dass die Oberfläche keine Nutzer verschreckt?), *Codometrics* (Woran kann man fehlerhaften und problematischen Code erkennen?) und *The Art of Readable Code* hatte ich mir letzteres angesehen. Es ist ein absolutes Einsteigerbuch, das man nicht mit den Klassikern wie dem *Clean Code* (besprochen in \$foo 2/2011) vergleichen kann. Denn es geht weniger auf größere Zusammenhänge ein, sondern beschränkt sich auf die Vermeidung der wesentlichen, leichter vermeidbaren Fallen. Diese werden aber knapp und unterhaltsam vorgestellt und sind für Aspiranten immer noch genügend Stoff, der ein bis zwei Jahre zur Verinnerlichung benötigt. Dank der Kürze und Anschaulichkeit, die mit einfacher Sprache, deutlichen Beispielen und eingeflochtenen Comicstreifen erreicht wurde, ist es durchaus ein Buch, das noch fehlte und Anfängern



dringend anzutragen ist. Trotzdem sollten Perl-Programmierer eher zu *Perl Best Practices* greifen, weil es viele der Ratschläge auch und noch einige mehr beinhaltet und speziell auf die vielen Eigenheiten von Perl eingeht. In diesem Werk sind die Beispiele in Python, C++, Java und Javascript geschrieben.

Die Rezension der Perl-Bibel schlechthin wird verschoben, da dieses Heft wahrscheinlich noch vor der fünften Edition von *Programming Perl* erscheint.

## Taschenreferenz

Kaum zu glauben, aber von der Seite gesehen, ist es vielleicht das älteste Perlbuch [3] überhaupt. Bereits um die Zeit der Version 2.0 (Ende 1988, das Kamel kam drei Jahre später) erschien im Netz der dreispaltige Spickzettel, aus dem sich die kompakte Übersicht des Sprachkerns entwickelte, die endlich 1996 in Buchform gebracht wurde. Sie wurde alle zwei Jahre aktualisiert, was allerdings vor neun Jahren einschloß. Das lag jedoch nicht daran, dass der alte Hase Johan sein Interesse an Perl verlor, sondern dass er seine Aufgabe sehr ernst nimmt. Während der weitreichenden Recherche zu dieser fünften Auflage gab es lange Diskussionen in der Mailingliste der Perlentwickler `p5p`, die zu Verbesserungen in den *perldoc*-Seiten führten. Denn Perl 5.10 und folgende brachten viel Neues, was noch nicht überall in der offiziellen Dokumentation ankam. Diese pflegt überdies manchmal einen Stil, die heute nur noch in kleineren Programmen vertretbar ist. Zum Glück ist vor allem brian d foy seit einiger Zeit damit beschäftigt, diese teilweise sehr kontroversen Probleme nach und nach mit 5.14, 5.14.1 und weiteren Versionen zu beheben.

Die Taschenreferenz ist ein wenig wortreicher und wesentlich umfangreicher verfasst als die `perldoc perlcheat`. Sie ist ein Stichwortgeber, den man aufsucht um die Schreibweise einer Spezialvariable oder magischen Konstante nachzuschlagen. Manchmal erinnert man sich nicht, ob der letzte Zugriff auf eine Datei das siebente oder achte Arrayelement von `stat` birgt (es ist Nummer acht). Dann ist es praktisch, dieses handtellergroße Büchlein neben dem Rechner liegen zu haben. Auch der Autor dieser Rezension lernte auf diese Weise wesentliche Feinheiten der Syntax. Neu hinzu kamen Tabellen zu den Funktionen des Smartmatch-Operators und

der IO-Layer, die neuen gierigen Quantoren, die experimentellen Steuersymbole fürs *Backtracking* usw. Die Linkliste musste sich sehr verändern und eine ganze Reihe Fehler oder fehlender Details, die sich all die Jahre und Auflagen hielten, wurde ausgebessert. Die wenig sagende Liste mit den Kernmodulen wurde gestrichen, wodurch die Seitenzahl sogar auf 93 nummerierte abnahm. Es ist die dünnste von allen Taschenreferenzen und ein paar Kleinigkeiten könnte man vermissen, wie die Rückgabewerte von `ref`. Mir fehlte leider auch die witzige Einleitung von Larry Wall, die nur in der Version für Perl 5.6 aus dem Jahr 2000 zu finden ist. Ebenfalls entschied sich Vromans strikt dagegen, irgendwelche Module zu beschreiben, so verbreitet und nützlich sie auch sein mögen (manche Kernmodule werden nur kurz angerissen). Er wünscht, dass weitere Referenzen zu Modulen oder Themen geschrieben werden. Ein passender Platz diese zusammenzutragen wäre das kürzlich gegründete <http://perl-tutorial.org/>. Die englische Ausgabe der Taschenreferenz erschien bereits im Juli und die deutsche wird wohl ab März 2012 erhältlich sein.

## Der Perl-Programmierer

Beinahe unbemerkt wurde im Sommer 2010 ein Perl-Buch in die Regale geschoben, dessen Konzept großes Potential birgt. Es will alles mitbringen, um aus einem Einsteiger einen kompletten Perl-Programmierer zu machen (daher der Titel). Jemanden, der die klassischen Skripte für Netzwerk, Mail und CGI im Halbschlaf schreiben kann. In drei Teilen lehrt es auf vielen Seiten die Grundlagen der Sprache, das Schreiben von Modulen und Objekten und die Anwendung der wichtigsten Module für den praktischen Einsatz. Dazu bietet es noch etwas Hintergrundinformationen zur Programmierung allgemein sowie zu den verwendeten Protokollen, die Professor Plate an der Universität München lehrt und noch genauer in anderen Büchern beschrieben hat.

Die reichlichen Beispiele lassen sich kopieren, erweitern und sofort nutzen. Es ist ein Nachschlagewerk für die praktische Arbeit. Darüber hinaus gibt es auf der Netzseite des Autors [4] noch einige Zusatzkapitel. Dabei ist auch eine kleine Taschenreferenz, die alle Funktionen und Kernmodule zusammenfasst. Die Liebe zu den Details zeigt sich bereits in den Zitaten, mit denen die Kapitel beginnen und die eines Damian Conway würdig wären. Eine Kostprobe:



*Where a calculator on the ENIAC is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and perhaps weigh 1.5 tons.*

*Popular Mechanics, März 1949*

Überhaupt wäre Jürgen Plate der richtige Mann dafür, einen unterhaltsamen Wälzer zu schreiben, der allen O'Reilly-Titeln trotzen könnte. Zu Recht schreibt der Hanser-Verlag über ihn: "Seit vielen, vielen Jahre ist er begeisterter Perl-User". Er hat ein Gefühl für die Perl-Kultur und gibt nicht viel auf oberflächliche Trends. Sein sachlicher, aber beschwingter Erzählton sucht keine Pointen, sondern kommentiert nur leicht ironisch, was die Wirklichkeit an Widersprüchen bietet. Das macht ihn teilweise angenehmer zu lesen als manchen der großen Namen. Dabei ist auch seine Lehrerfahrung spürbar. Kaum eine andere Perl-Fibel nimmt sich den Platz, die benutzten Technologien zu erklären. Der Anhang zeigt darüber hinaus, dass Plate die Literatur gut kennt.

Und es muss wirklich nicht immer ein Einband mit einem Tier sein, selbst wenn das Buch in diesem Haus unter *Andere Programmiersprachen* im Katalog eingeordnet ist. Hanser hat zum Beispiel mit Kernighans und Ritchies *Programmieren in C* selbst Klassiker im Angebot und die Reihe *Der ...-Programmierer* ist eine ernsthaft geführte Serie, deren C++-Ausgabe erst dieses Jahr mit der Erneuerung des Standards eine Neuauflage erfuhr.

Trotzdem ist dieses Werk nur eingeschränkt empfehlenswert. Zum Beispiel steht auf dem Umschlag, es sei *ein Praxisbuch für alle Ansprüche – mehr brauchen Einsteiger und Profis nicht*. "Die Perl-Programmierer" benutzen heute `Dist::Zilla`, `Catalyst`, `Mojolicious` oder `Dancer`, `Plack`, `Wx` oder `GTK`, `POE`, `AnyEvent` und vieles mehr. Nichts davon wird im Text erwähnt. Das ist allerdings nicht unbedingt ein Versäumnis, denn die inhaltliche Beschränkung entspricht grob einem verbreiteten Nutzerprofil und macht sehr viel Sinn. Nur hätte man es ehrlicher ausweisen sollen als "umfassende Anleitung für werdende sehr gute Programmierer, welche die meisten traditionellen Aufgaben mit Sicherheit lösen können".

Für mein Empfinden unterscheidet einen "Programmierer" von einem "Skripte-Schreiber", dass er auch mit größeren Strukturen und Projekten umgehen kann. Themen wie Dokumentation, Objekte, das Testen und der Debugger wurden dazu behandelt, Ausnahmebehandlung, Profiling und Benchmarking allerdings nicht. Dafür wurden angenehm viele Perl-Spezifika wie `AUTOLOAD`, `Taint`, `Typeglobs` und höhere `Regex`-Funktionen vorgestellt, seltsamerweise nicht `Tie`. Insgesamt wurde großer Wert darauf gelegt, jeden scheinbar unbedeutenden Zwischenschritt (Wie schreibe ich eine `README`?) zu erläutern.

Wirklich schmerzlich war jedoch die Empfehlung von `Switch`, was nach langen Ankündigungen und Warnungen vor den Nebenwirkungen mit Erscheinen des Buches aus dem Perl-Kern entfernt wurde.

`given/when` wurde nur im Nebensatz erwähnt, seine vielen praktischen Möglichkeiten blieben verschwiegen. Das gilt für alles, was mit Version 5.10 kam (siehe `$foo`-Ausgaben 1 und 2 2007). Da "Der Perl-Programmierer" zweieinhalb Jahre später veröffentlicht wurde, ist das kaum entschuldbar. Auch `Moose`, `Try::Tiny`, `autodie`, `Perl::Critic` und vieles mehr waren 2010 kein Geheimtipp mehr. In vielen weiteren Details merkt man, dass Herr Plate nicht am Puls der Entwicklung lebt. (Selbst bei einer Entscheidung für `Tk` gehört unbedingt `Tkx` dazu.) Täte er das, wäre es mit ein wenig optischer Politur ein rundum gutes, vielleicht sogar maßgebliches Buch für Anfänger bis weit Fortgeschrittene geworden. Leider ist eine Neuauflage derzeit nicht geplant.

#### Links

- [1] <http://www.oreilly.de/catalog/>
- [2] <http://www.hanser.de/nb.asp?area=Computer>
- [3] <http://www.vromans.org/johan/perlref-history.html>
- [4] <http://www.netzmafia.de/skripten/buecher/index.html>