

Vorwort

*In my eyes it is never a crime to steal knowledge. It is a good theft.
The pirate of knowledge is a good pirate.
Michel Serres*

Ein kurzes Interview

Worum geht es in diesem Buch? *Clemens:* Wir wollen den Lesern helfen, den Programmieralltag mit Ruby zu bewältigen.

Armin: Dafür bieten wir eine praxisorientierte Einführung in die objektorientierte Programmiersprache Ruby. Zusätzlich haben wir Schnelleinführungen für erfahrene Perl-, Java- oder C++-Programmierer, die neue Jagdgründe erobern wollen.

Warum habt ihr das Buch geschrieben? *Clemens:* Um Programmierern Ruby näher zu bringen und sie dadurch im sonst grauen Programmieralltag »glücklich« zu machen.

Stefan: Wir wollten die Einführung in das Programmieren mit Ruby schreiben, die wir selbst gebraucht hätten, als wir Ruby gelernt haben.

Armin: Ruby hat uns von sich überzeugt. Bei Approximity setzen wir Ruby für die verschiedensten Aufgaben ein: von Wegwerfskripten über Prototyping bis zu größeren Analyse- und Bestellsystemen in Intranets.

Für wen ist das Buch? *Armin:* Für alle, die effizient programmieren wollen. Nach unseren Erfahrungen lieben vor allem Smalltalk-Fans Ruby, aber auch Java-Programmierer kommen schnell auf den Geschmack.

Clemens: Programmierer, die sich der Mächtigkeit von Skriptsprachen bewusst sind und diese gepaart mit Eleganz in der Praxis einsetzen wollen. »Aufstieg« für Tcl-, Perl- und Python-Programmierer, »Umstieg« für Smalltalk-, Java- und C++-Programmierer.

*Siehe Geleitwort von
matz*

i -)

Für wen ist dieses Buch nicht geeignet? *Stefan:* Für Leser, die eine vollständige und endgültige Referenz erwarten. Ruby und seine Bibliotheken sind nach wie vor im Wachsen, und in jeder Release gibt es ein paar neue Features, die das Leben leichter machen.

Wir gehen davon aus, dass unsere Leser Ruby lernen wollen, und sind davon überzeugt, dass der beste Weg dazu ist, mit der Sprache zu experimentieren. Wer erwartet, nur durch Lesen dieses Buches die Sprache zu beherrschen, wird nicht zufrieden sein.

Welches Vorwissen sollte man mitbringen? *Clemens:* Programmiererfahrung in einer vorzugsweise objektorientierten Sprache ist vorteilhaft. Eine kurze Einführung geben wir in diesem Buch im Kapitel 15.1.

Armin: Erfahrungen mit einer Skriptsprache wie Perl oder PHP schaden nicht, denn Ruby ist Perl relativ ähnlich.

Wie ist das Buch aufgebaut? Was steht in den einzelnen Teilen und Kapiteln? *Stefan:* Das Buch ist in 5 Teile gegliedert:

1. Ruby kennen lernen: Bei diesem ersten Rendezvous zeigen wir einige Ruby-Codefragmente, die »typisch Ruby« sind, und wie man sie ausprobieren kann.
2. Die Einführung in Ruby enthält das »alltägliche Handwerkszeug«, von Zahlen und Strings über Schleifen, Arrays und Hashtabellen bis zu Iteratoren, Continuations und Threads.
3. Objektorientiertes Programmieren in Ruby: Wir bieten einen kurzen Einstieg in objektorientiertes Arbeiten mit Ruby und erklären, wie man mit Objekten, Klassen und Modulen umgeht. Dieser Teil schließt mit einem Ausblick auf Design Patterns (Entwurfsmuster), Metaprogrammierung und Reflexion.
4. Im »Kochbuch« geht es um Ruby-Anwendungen, die im »täglichen Leben« eines Programmierers wichtig sind: CGI, XML, SOAP, Datenbank, grafische Oberflächen und noch mehr . . .
5. Dann haben wir noch den Anhang mit Installationshinweisen, »Schnellkursen« für Umsteiger, einer Bibliographie und einem Stichwortverzeichnis.

*Ruby-Chan zeigt hier
ihre Eleganz und
Schönheit!*

Wie sollten Einsteiger das Buch lesen? *Armin:* Ich würde Ruby gemäß Anhang E installieren, mir dann die »Schmankerl« anschauen und das Buch mit Randbemerkungen und Zitaten kurz überfliegen. Danach würde ich es Seite für Seite durcharbeiten, die Beispiele am PC ausprobieren und vielleicht auch gleich ein bisschen abändern. Learning by doing ist effizienter als sich berieseln zu lassen. Wir haben deswegen auch unsere Programmbeispiele bewusst sparsam kommentiert.

Siehe Abb. 1

Stefan: Wer noch keine Erfahrungen mit objektorientierten Sprachen hat, möge bitte die »besondere« Schreibweise der Ruby-Methoden als syntaktische Regel akzeptieren. Mit zunehmender Erfahrung kommt dann auch das Verständnis, was eigentlich passiert.

Clemens: Wir haben einige Stellen im Text mit einem Rubin gekennzeichnet, die man beim ersten Durchlesen getrost überspringen kann, da sie fortgeschrittene Themen ansprechen oder Verweise auf nachfolgende Kapitel enthalten.

Wie sollten Umsteiger und Fortgeschrittene das Buch lesen? *Clemens:* Für Umsteiger von Perl, Java und C++ haben wir im Anhang Kurzinformationen zusammen gestellt, die wir vor allem in der Ruby-Talk-Mailingliste gesammelt haben.

Siehe Abb. 1

Stefan: Außerdem weisen wir im Text immer wieder darauf hin, wie bestimmte Features in Ruby mit anderen Sprachen in Beziehung stehen.


Was kann der Leser, wenn er das Buch durchgearbeitet hat? *Stefan:* Er weiß, wofür Ruby eingesetzt werden kann und kennt den für Ruby typischen Programmierstil.

Armin: Er programmiert produktiv in einer modernen, objektorientierten Sprache. Return on Investment ist garantiert!

Was muss man sonst noch über dieses Buch wissen? Dieses Buch erhebt nicht den Anspruch eine vollständige Referenz zu sein, wie sie mit dem exzellenten »Programming Ruby« der Pragmatic Programmers [ThHu99] schon (auf Englisch und wohl bald auch auf Deutsch) zur Verfügung steht. Deshalb schadet es nicht, ein Exemplar des so genannten »Pickaxe«-Buchs zur Hand zu haben, das auch als kostenlose HTML-Version [PickAxe] oder als HTML-Hilfe-Datei im Windows-Installer [WinRuby] verfügbar ist. Und kurz vor Weihnachten 2001 ist uns noch die Referenz »Ruby in a

Nutshell« [Mat01] von matz selbst ins Haus geflattert, die auch Änderungen bis zur Version 1.7.1 erfasst.



Tiefer gehende Konzepte und Anmerkungen, die beim ersten Lesen übergangen werden können, haben wir mit dem Maskottchen Ruby-Chan markiert und in einer anderen Schrift gesetzt. 

Ruby-Programme sind in dieser Schrift gesetzt, Ergebnisse, die an der Konsole ausgegeben oder in der Ruby-Shell `irb` angezeigt werden, markieren wir mit `#->`.

Abbildung 1
Das Buch im
Überblick

```
Appetithappen.anschauen

case Programmierer
  when PerlFan then Anhang['PerlUmsteiger'].lesen
  when JavaFan then Anhang['JavaUmsteiger'].lesen
  when CplusplusFan then Anhang['CplusplusUmsteiger'].lesen
  when PythonFan
    RubyPython.lesen
    Teil['II'].lesen
    Teil['III'].lesen
  when Neuling
    Beispiele.testenAuf('www.ruby.ch')
    Anhang['RubyInstallieren'].lesen
    Teil['II'].lesen
    Teil['III'].lesen
    Projekte.verstehen
  when Problemsuche
    Inhaltsverzeichnis.durchsehen
    Index.durchsehen
  when RubyGuru
    ['III', 'IV', 'V'].each { | nr |
      Teil[nr].lesen
    }
  when Ich_will_Code_sehen then Projekte.anschauen
  when absolut_verloren
    Buch.lesen
    Ferien.machen
  when gelangweilt then Zitate.anschauen
end
```

Danksagungen der Autoren

An der Entstehung dieses Buches waren direkt und indirekt viele Menschen beteiligt:

- ❑ matz, der die letzten 7 Jahre in Ruby investiert hat.
- ❑ Teilnehmer an Mailinglisten (ruby-talk, ruby-de), vor allem Guy Decoux, Andy Hunt, Dave Thomas, David Alan Black, Guy H. Hurst, Michael Neumann, Conrad Schneiker, Clemens Hintze und Robert Feldt, um nur einige zu nennen.
- ❑ Andrew Hunt und Dave Thomas mit ihrem »Pickaxe«-Buch: Programming Ruby [ThHu01], das auch online [PickAxe] zur Verfügung steht.
- ❑ Die Bibliotheksprogrammierer, unter ihnen: Shugo Maeda (eruby, mod_ruby), Avi Bryant (Iowa), Keiju Ishitsuka (irb), Bryan Zarnett (Facet), Takahashi Masayoshi (sablot), Michael Neumann (DBI, XML-RPC), Tomita Masahiro (MySQL), Jim Menard (NQXML), Hiroshi Igarashi (GTK), Lyle Johnson (FXRuby), Yoshida Kazuhiro (Apollo), Goto Kentaro, Masaki Suketa (RubyUnit), Yuichi Takahashi (WebUnit), Hiroshi Yokota (DNS Balance), Emil Ong (Ruby-MPI).
- ❑ Unermüdliche Fragenbeantworter, wie zum Beispiel Kiriya-ma Kazuhiko (Perl und Python für Ruby).
- ❑ Unsere Reviewern, allen voran Clemens Hintze, die gnadenlos die Schwächen unseres Manuskripts ausfindig gemacht haben. Ursular Zimpfer hat bei der Rechtschreibkorrektur Erstaunliches geleistet.
- ❑ Clemens Hintze und Michael Neumann, die uns nach ihrer Review jeweils einen Abschnitt über ein Spezialthema spendiert haben.
- ❑ René Schönfeldt und seine Kollegen vom dpunkt.verlag, die sich mit einem jungen Autorenteam viel Mühe gegeben haben.

Do you ever sleep?

Sometimes. matz

Inbesondere bedanken wir uns bei

- ❑ der Gaby, der »Lady«, der »Zwetschgenlise« und dem »Tiger« für deren Verständnis für Clemens' »autorenbedingten« Abwesenheiten ;-)

- ❑ Isabella, Herta, Kurt, Ingrid und Karl-Heinz, die Stefan unterstützen, wenn es zählt, sowie bei Helena und Alexander dafür, dass Stefan ab und zu seinen Rechner benutzen durfte,
- ❑ dem Proximity-Team sowie à tous les sales alligators!

Internet-Ressourcen

Ruby entwickelt sich stetig weiter. Möchte man Ruby effizient einsetzen, sollte man auf dem Laufenden bleiben. Dazu finden sich im Internet mehrere Möglichkeiten:

www.ruby-lang.org »matz«: Die Originalsite

www.ruby-lang.org/en/raa.html *viele Autoren*: Ruby Application Archive

www.ruby-lang.org/en/doc.html *viele Autoren*: aktuelle Ruby-Dokumentation

www.ruby.ch *Clemens Wyss*: Der Online-Ruby-Interpreter

www.rubycentral.com *Dave Thomas, Andrew Hunt*: Windows-Installer, Referenzmaterial, »Programming Ruby« zum Herunterladen

www.ale.cx/mine *Alexi Niemela*: Das RAA (Ruby Application Archive) schön »aufbereitet«

www.rubygarden.org *viele Autoren*: News rund um Ruby, Ruby Change Requests, ein WikiWiki

homepage1.nifty.com/markey/index_e.html *Suketa' Zakki-Cho*: RubyWin, win32ole, RubyUnit

Die besten Informationsquellen sind nach wie vor die Mailinglisten. Auf www.ruby-talk.org liegt das Archiv der englisch-sprachigen ruby-talk-Liste, allgemeine Informationen zu den aktuell aktiven Listen finden sich unter <http://www.ruby-lang.org/en/ml.html>.