

# [PDF] Python Essential Reference (3rd Edition)

David M. Beazley - pdf download free book

---

**Books Details:**

Title: Python Essential Reference (3

Author: David M. Beazley

Released: 2006-03-02

Language:

Pages: 648

ISBN: 0672328623

ISBN13: 978-0672328626

ASIN: 0672328623



**[CLICK HERE FOR DOWNLOAD](#)**

---

pdf, mobi, epub, azw, kindle

## Description:

Every so often a book comes along that makes you ask yourself, "Gee, when was the last time I had my eyes checked?" David M. Beazley's *Python: Essential Reference* is just such a book. Condensing thousands of pages of Python online documentation into a compact 319-page softcover, Beazley and his editors used the old-college trick (often performed in reverse) of dickering with the font size to meet a putative page-limit requirement. The result is a truly condensed product fit for the occularly well-adjusted (nota bene).

Beazley's subject is Python, a full-featured, freely-redistributable, POSIX-compliant (platforms include Linux, Unix, Macintosh, and Windows) scripting language that is based on object-oriented design principles. As advertised, Beazley's source release (1.5.2) is available from an unfortunately

slow server at [www.python.org](http://www.python.org). The installation under Linux (Redhat 5.2) proceeded without incident.

Beazley holds true to his catalogic purpose: fully 230 pages are formatted as technical appendices and indices covering the standard litany: built-in function syntax, database features, OS-level interfaces, Internet interfaces, and compiling/profiling/debugging. All references are fully annotated and illustrated with example source code that runs from a couple of lines to a couple of pages. In lock step with competing scripting languages, Python is extensible and embeddable in C and C++, and with blitzkrieg efficiency, Beazley summarizes these crucial practical issues in the final 30 pages. Python users who are tired of chasing questions through hyperlinked online documents will benefit from the expansive random-access index.

*Python* the book captures the orderliness of Python the language. Beazley begins with an 86-page précis of Python in the fashion of Kernighan and Ritchie: too brief for a newbie tutorial but enough to propel old hands into a scripting language that aspires to the elegance of a compiled language.

Indeed, it is a byte-compiling language. The line `bytecode=compile("some_python_script","exec')` creates 'bytecode' as a token executed by `exec bytecode`. But a five-minute investigation through Beazley's book does not describe how 'bytecode' can be written into a separate executable file. If writing the byte-compiled code to a file is not possible, Python suffers from the limitations of other scripting languages: the executable is the source and cannot be hidden from the user, at least not without some difficulty. Despite its extensibility, embeddability, and pleasing architecture, Python is like other scripting languages: appropriate for solving small nonproprietary problems.

Those familiar with more established scriptors like Perl may ask, "Why Python?" Unlike Perl, Python is a product of the fully object-oriented (OO) era, and its constructs reflect design principles that aspire beyond keystroke shortcuts of the succinct-but-often-arcane Perl. Python creator Guido van Rossum cleansed Perl's idiosyncracies and objectified basic data structure, data manipulations, and I/O. With Python, OO is so intrinsic that learning Python is equivalent to learning OO. The same cannot be said of Perl.

Unfortunately, comparisons with other languages are missing from Beazley's book. Van Rossum, in an embarrassingly self-serving foreword, preemptively asserts that we readers need "neither evangelizing nor proselytizing"--after all, we already own the book--but we do need galvanizing and we don't find it. Specifically, we need a response to the oft-repeated wisdom that new computer languages are only worth learning if they teach us to organize our thinking along new lines.

Scripting languages, however, are for quick and dirty projects: quick to write, easy to hack, and ultimately disposable. The essential tension created by van Rossum and friends is between the elegance of object-oriented principles and the utility of a quick-hacked script. Sadly, the tension remains unresolved in Beazley's reference. There is little to convince us that Python has earned its place in the firmament by changing our thinking. But Beazley has given us much to get us going if we have already taken the leap of faith. --*Peter Leopold* --This text refers to an out of print or unavailable edition of this title.

**From Library Journal** Though Python is a relatively new programming language, it has quite a significant audience owing to its sensible syntax. An active user of Python since 1996, Beazley provides ample information on the fundamentals of versions 2.0 and 2.1, including syntax, functions, operators, classes, and libraries. This is first and foremost a reference, so he avoids lengthy discussions of Python's superiority. Peppered with good code samples and featuring a companion web site with more extensive pieces, this title should be on hand in larger libraries.

Copyright 2001 Reed Business Information, Inc. --This text refers to an out of print or unavailable

edition of this title.

---

- Title: Python Essential Reference (3rd Edition)
  - Author: David M. Beazley
  - Released: 2006-03-02
  - Language:
  - Pages: 648
  - ISBN: 0672328623
  - ISBN13: 978-0672328626
  - ASIN: 0672328623
-