Published on *InfoWorld* (http://www.infoworld.com)

Pillars of Python: Six Python Web frameworks compared

By Rick Grehan
Created *2011-08-10 03:00AM*

Although Python [1] is not as prevalent as, say, PHP [2] as a language for Web applications [3], Python nevertheless has much to recommend it in that effort. It is a dynamic, interpreted language, as is PHP, and therefore encourages iterative, exploratory development. Programming purists could point to the fact that object orientation was designed into Python from its very beginnings, rather than being retrofitted to the language at a later point in its life.

Language design considerations aside, the capabilities embodied in Python's standard libraries are impressive. Python even installs with its own Web server. In addition, Python boasts plenty of free database libraries, numerous free Web page template systems, and even libraries for interfacing with your favorite Web server, all ready to download and apply to your next Web application project.

[Also on InfoWorld: "InfoWorld review: Nine fine Python development tools [4]" | Neil McAllister reveals the most dangerous programming mistakes [5]. | Get software development news and insights from InfoWorld's Developer World newsletter [6].]

Of course, you don't need to do all that work. The engineers of these capable and diverse Python-based Web frameworks have done it all for you.

In the following pages, we review six Web application frameworks for the Python Web developer. These are by no means the only Python Web frameworks available, but represent a broad sampling of the possibilities. No matter what your needs or leanings as a Python developer might be, one of these frameworks promises to be a good fit.

They include Zope 2 [7], the proud descendant of one of the longest-established Web server frameworks in any language; Web.py [8], a low-abstraction framework that provides only the essentials for Web development; Web2py [9], a higher-level framework that provides its own Web-based integrated development environment; Pyramid [10], a flexible new offering from the group that produced the popular Pylons framework; the popular and highly regarded Django [11] framework; and CubicWeb [12], with which you can build not just Web applications but semantic Web applications.

Python Web frameworks: Light, heavy, and in between

Zope 2 [7] is a descendant of Zope, the great-granddaddy of all Python Web application tools and an important branch in the Python family tree. Zope 2's ancestry, then, goes back to the mid-1990s, and Zope's continued popularity is testament to its solidity. Consequently, one huge advantage to choosing Zope 2 is that you can call on 15 years of code development and documentation.

Zope 2's "object publishing" system fits well into an object-oriented development mind-set and mitigates somewhat the learning curve you'd have to climb if you picked a framework that exposes your application to more of the more painful aspects of Web development. Zope 2 does present a formidable learning curve of its own -- but you'll at least enjoy the stability that a 15-year pedigree confers.

The Web2py [9] framework is an abstraction paradise. Databases hide behind a Database Abstraction Layer (DAL). Web2py's rendering system will try to find a view that -- depending on context -- displays an object in HTML, XML, JSON, or any of the half-dozen protocols the framework supports. Intelligently crafted by a professor of computer science, Web2py's template system actually lets you use Python as an HTML-embeddable scripting language. Falling somewhere between the all-encompassing world of Zope 2 and the minimalism of Pyramid, Web2py may well be the best framework for Python-savvy developers to enter the world of framework-based Web application development.

The philosophy of Web.py [13] -- a minimalist framework -- is not to abstract away the details of interacting with the Web, but to make that interaction easier. As a result, you'll find yourself writing HTTP GET function handlers directly. Likewise, the Web.py database system does not abstract away SQL; rather than hide the fact that you're querying a database, it hides the details of working with different databases. Web.py does define a template language, which -- like that of Web2py -- lets you embed arbitrary Python code in a Web page. Web.py is ideal if you're already familiar with building Web applications (perhaps you once wrote CGI-based applications). You'll get started quickly with Web.py, but you'll have to rely on your own wits to go beyond simple Web applications.

Pyramid [10] is also minimalist Web framework, not so much in its capability as its philosophy. It makes no assertion concerning the back-end database you should use, nor does it foist a particular template system on the developer. (Currently, Pyramid supports two, though Pyramid itself tries to remain agnostic regarding the choice.) If Pyramid has any blemishes, it is the quirky terminology with which its documentation describes the framework. In addition, its laissez-faire attitude can leave you wondering precisely how to proceed to accomplish a particular task -- and, when you manage to accomplish that task, whether your solution was the best. For example, there are two separate means of determining how your application will handle a given URL: the well-known URL-mapping mechanism and something called "traversal" (which you will need to read the documentation to figure out). So, with Pyramid, you purchase flexibility at the cost of turning yourself into something of an explorer.

Django [11] is a mature and highly regarded Web framework that assumes you know Python well. Its libraries provide a good selection of the must-have capabilities for accelerating Web application development: an object-relational mapper so that you don't have to write your own RDBMS interface code, a template system so that you don't have to wrestle with marrying active Python content to static HTML content on a Web page, an administrative interface so that you can easily access your site's back-end data (as well as easily manage website users and permissions), and so on. You should expect to spend a stretch learning Django's API, but the time you put into Django is well worth it. It's a rich environment with much to recommend it.

CubicWeb [12] touts itself as not merely a Web development framework, but a semantic Web development framework. This distinction becomes clear when you discover that a CubicWeb application's interaction with a database is performed through RQL (Relation Query Language), a query language similar to the W3C's SPARQL for RDF. CubicWeb's libraries translate RQL queries to SQL or XML or LDAP or whatever protocol is appropriate for the data store being accessed. CubicWeb is probably the most difficult framework in this review to grasp. The difficulty is not only on account of the jargon you must learn, but also because constructing a CubicWeb application entails lashing together modules (called cubes) into a final structure -- a process that will be foreign to neophytes. Once you get the hang of it, however, CubicWeb does permit rapid development. A basic model schema is enough to get an application off the ground, and CubicWeb's agile characteristics let you grow the application in iterative fashion.

Python Web frameworks: Min or max?

If you prefer a framework that puts the minimum between you and the Web, then Web.py [13] will be your best choice. On the other hand, if you like wizards guiding you along the way, then you may prefer Web2py [9]. CubicWeb [12] is an excellent choice if your data comes from disparate sources. If your website's structure is data driven, then have a look at Pyramid [10] or Zope 2 [7]. In addition, Zope 2 [7] and Django [11] are solid all-around choices, both having stood the test of time.

But these are only general suggestions. It's not the case that any particular Python Web framework is at a significant disadvantage to the others. As usual, the choice is highly subjective. You will find zealots for each product, and every zealot is able to present rational reasons why their chosen framework is superior.

Naturally, the current discussion paints only an overview. For the finer details, follow the links in the table below and plunge into the individual reviews.

Python Web frameworks compared

	Python versions	Licensing	Documentation	Web servers
CubicWeb 3.12.5 [12]	Any 2.x Python back to 2.5	LGPL	Tutorials, online user guide, and administration guide, all available from the main Web page	Twisted Web server, usually run with Apache. WSGI support is in the works.
Django 1.3 [11]	Any 2.x Python version	BSD	Tutorial, reference guide, and how-to guides, arranged like an online book	Any Web server that supports WSGI or FastCGI
Pyramid 1.0 [10]	Python 2.4 through 2.7	BSD-like	User guide, tutorials, sample applications, and API reference	Any Web server that supports WSGI. Also includes its own Web server suitable for large-scale applications.
		No restrictions,		

Web.py 0.35 [13]	Any 2.x Python back to 2.3	but includes the CherryPy Web server governed by the CherryPy license	Cookbook, API reference, and categorized code examples	Any Web server supporting CGI, FastCGI, SCGI, or WSGI
Web2py 1.95 [9]	Python 2.5 through 2.7	LGPL	Online book, example website, lots of online "quick examples"	Any Web server that supports Python
Zope 2.13 [7]	Depends on Zope 2 version, varying from Python 2.4 for Zope 2.11.x to Python 2.7 for Zope 2.13.x	Zope Public License [14]	Online book includes API reference, management interface guide, templates reference, and more	Any Web server that supports Python. Also includes its own Web server suitable for production use.

Python Web frameworks compared, continued

	Database support	Caching	Debugging	Logging	JavaScript support
CubicWeb 3.12.5 [12]	MySQL, PostgreSQL, SQLite, Microsoft SQL Server	Planned	Debug mode will display log messages on a console, show trace information, show generated SQL queries, and more	Yes	Yes, based on JQuery
Django 1.3 [11]	MySQL, PostgreSQL, SQLite, Oracle; third-party support for other databases	Yes, both at the model and view levels	Yes; both in its development Web server and a special debug execution mode	Uses standard Python logging; latest version added configuration hooks	None specific, though third-party libraries for integrating JavaScript libraries are available
Pyramid 1.0 [10]	Any Python ORM tool such as SQLAlchemy, MongoDB, and Cassandra	Yes, via the Python Beaker package; provides page-level database query and other	Yes; interactive debugger that runs in the browser provides detailed stack trace	Yes, via standard Python logging package	No, though Pyramid does provide convenience functions for JSON and Ajax

		caching			
Web.py 0.35 [13]	MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle, Firebird	Some support for caching templates	Yes; debug mode will automatically load code changes and provide detailed error pages	No, but you can incorporate the Python logging system	Yes, via jsdef (used in the template language)
Web2py 1.95 [9]	MySQL, PostgreSQL, SQLite, Microsoft SQL Server, Oracle, IBM DB2, Informix, Ingres, Firebird, Google App Engine	Yes; caching is integrated with database queries	No, but logging system tickets any uncaught exceptions	Yes, using the standard Python logging module; every app writes to its own log	Yes, custom JavaScript atop JQuery is used in various places
Zope 2.13 [7]	Zope Object Database (ZODB) and third-party ZODB emulators such as RelStorage	Yes; includes several built-in caching helpers	Yes; debug mode lets you run Zope while watching debugging and logging output	Yes; uses standard Python logging capability	None specific

This article, "Pillars of Python: Six Python Web frameworks compared [15]," was originally published at InfoWorld.com [16]. Follow the latest developments in application development [17] and Python [18] at InfoWorld.com. For the latest developments in business technology news, follow InfoWorld.com on Twitter [19].

Application Development Open Source Software Application Development Python Web Development

Links:
[1] http://www.infoworld.com/t/python
[2] http://www.infoworld.com/d/developer-world/infoworld-review-fabulous-php-frameworks-364
[3] http://www.infoworld.com/t/web-applications
[4] http://www.infoworld.com/d/developer-world/infoworld-review-nine-fine-python-development-tools-374?source=fssr
[5] http://www.infoworld.com/d/application-development/developer-error-the-most-dangerous-programming-mistakes-706?source=fssr
[6] http://www.infoworld.com/newsletters/subscribe?showlist=infoworld_developer&source=ifwelg_fssr
[7] http://www.infoworld.com/d/application-development/pillars-python-zope-2-web-framework-168935
[8] http://www.infoworld.com/d/application-development/programming-python-webpy-web-framework-169072
[9] http://www.infoworld.com/d/application-development/pillars-python-web2py-web-framework-168920
[10] http://www.infoworld.com/d/application-development/pillars-python-pyramid-web-framework-168661

[11] http://www.infoworld.com/d/application-development/pillars-python-django-web-framework-168643
[12] http://www.infoworld.com/d/application-development/pillars-python-cubicweb-web-framework-169105
[13] http://www.infoworld.com/d/application-development/pillars-python-webpy-web-framework-169072
[14] http://www.zope.org/Resources/License
[15] http://www.infoworld.com/d/application-development/pillars-python-six-python-web-frameworks-compared-169442?source=footer
[16] http://www.infoworld.com/?source=footer
[17] http://www.infoworld.com/d/application-development?source=footer
[18] http://www.infoworld.com/t/python?source=footer
[19] http://twitter.com/infoworld