

PyYAML

PyYAML is a YAML parser and emitter for Python.

Overview

YAML is a data serialization format designed for human readability and interaction with scripting languages.

PyYAML is a YAML parser and emitter for the Python programming language.

PyYAML features

- a **complete** **YAML 1.1** parser. In particular, PyYAML can parse all examples from the specification. The parsing algorithm is simple enough to be a reference for YAML parser implementors.
- Unicode support including UTF-8/UTF-16 input/output and `\u` escape sequences.
- low-level event-based parser and emitter API (like SAX).
- high-level API for serializing and deserializing native Python objects (like DOM or pickle).
- support for all types from the **YAML types repository**. A simple extension API is provided.
- both pure-Python and fast **LibYAML**-based parsers and emitters.
- relatively sensible error messages.

Requirements

PyYAML requires Python 2.5 or higher. PyYAML also supports Python 3.

Download and Installation

The current stable release of PyYAML: **3.11**.

Download links:

- **TAR.GZ package:** <http://pyyaml.org/download/pyyaml/PyYAML-3.11.tar.gz>
- **ZIP package:** <http://pyyaml.org/download/pyyaml/PyYAML-3.11.zip>
- **Windows installers (32-bit):**
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py2.6.exe> (for Python 2.6)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py2.7.exe> (for Python 2.7)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py3.1.exe> (for Python 3.1)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py3.2.exe> (for Python 3.2)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py3.3.exe> (for Python 3.3)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win32-py3.4.exe> (for Python 3.4)
- **Windows installers (64-bit):**
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py2.6.exe> (for Python 2.6)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py2.7.exe> (for Python 2.7)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py3.1.exe> (for Python 3.1)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py3.2.exe> (for Python 3.2)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py3.3.exe> (for Python 3.3)
 - <http://pyyaml.org/download/pyyaml/PyYAML-3.11.win-amd64-py3.4.exe> (for Python 3.4)

Unpack the archive and install the package by executing

```
$ python setup.py install
```

If you want to use [LibYAML](#) bindings, you need to download and install [LibYAML](#). Then you may install the bindings by executing

```
$ python setup.py --with-libyaml install
```

The source distribution includes a comprehensive test suite. To run the tests, type

```
$ python setup.py test
```

You may check out the latest development code of PyYAML from the Subversion repository <http://svn.pyyaml.org/pyyaml/trunk>

```
$ svn checkout http://svn.pyyaml.org/pyyaml/trunk pyyaml-trunk
```

Documentation

Quick example (see documentation for loading multiple documents):

```
>>> import yaml

>>> print yaml.load("""
... name: Vorlin Laruknuzum
... sex: Male
... class: Priest
... title: Acolyte
... hp: [32, 71]
... sp: [1, 13]
... gold: 423
... inventory:
... - a Holy Book of Prayers (Words of Wisdom)
... - an Azure Potion of Cure Light Wounds
... - a Silver Wand of Wonder
... """)

{'name': 'Vorlin Laruknuzum', 'gold': 423, 'title': 'Acolyte', 'hp': [32, 71],
'sp': [1, 13], 'sex': 'Male', 'inventory': ['a Holy Book of Prayers (Words of Wisdom)',
'an Azure Potion of Cure Light Wounds', 'a Silver Wand of Wonder'], 'class': 'Priest'}

>>> print yaml.dump({'name': "The Cloak 'Colluin'", 'depth': 5, 'rarity': 45,
... 'weight': 10, 'cost': 50000, 'flags': ['INT', 'WIS', 'SPEED', 'STEALTH']})

name: The Cloak 'Colluin'
rarity: 45
flags: [INT, WIS, SPEED, STEALTH]
weight: 10
cost: 50000
depth: 5
```

For more details, please check [PyYAML Documentation](#).

History

3.11 (2014-03-26)

- Source and binary distributions are rebuilt against the latest versions of Cython and

LibYAML.

3.10 (2011-05-30)

- Do not try to build LibYAML bindings on platforms other than CPython; this fixed installation under Jython (Thank to olt(at)bogosoftware(dot)com).
- Clear cyclic references in the parser and the emitter (Thank to kristjan(at)ccpgames(dot)com).
- LibYAML bindings are rebuilt with the latest version of Cython.
- Dropped support for Python 2.3 and 2.4; currently supported versions are 2.5 to 3.2.

3.09 (2009-08-31)

- Fixed use of uninitialized memory when emitting anchors with LibYAML bindings (Thank to cegner(at)yahoo-inc(dot)com).
- Fixed emitting incorrect BOM characters for UTF-16 (Thank to Valentin Nechayev)
- Fixed the emitter for folded scalars not respecting the preferred line width (Thank to Ingy).
- Fixed a subtle ordering issue with emitting %TAG directives (Thank to Andrey Somov).
- Fixed performance regression with LibYAML bindings.

3.08 (2008-12-31)

- Python 3 support (Thank to Erick Tryzelaar).
- Use Cython instead of Pyrex to build LibYAML bindings.
- Refactored support for unicode and byte input/output streams.

3.07 (2008-12-29)

- The emitter learned to use an optional indentation indicator for block scalar; thus scalars with leading whitespaces could now be represented in a literal or folded style.
- The test suite is now included in the source distribution. To run the tests, type python setup.py test.
- Refactored the test suite: dropped unittest in favor of a custom test appliance.
- Fixed the path resolver in CDumper.
- Forced an explicit document end indicator when there is a possibility of parsing ambiguity.
- More setup.py improvements: the package should be usable when any combination of setuptools, Pyrex and LibYAML is installed.
- Windows binary packages are built against LibYAML-0.1.2.
- Minor typos and corrections. Thank to Ingy dot Net and Andrey Somov.

3.06 (2008-10-03)

- setup.py checks whether LibYAML is installed and if so, builds and installs LibYAML bindings. To force or disable installation of LibYAML bindings, use --with-libyaml or --without-libyaml respectively (partially fixes #34).
- Building LibYAML bindings no longer requires Pyrex installed (fixed #33).
- yaml.load() raises an exception if the input stream contains more than one YAML document (fixed #54).
- Fixed exceptions produced by LibYAML bindings (fixed #50).
- Fixed a dot '.' character being recognized as !!float (fixed #62).
- Fixed Python 2.3 compatibility issue in constructing !!timestamp values.
- Windows binary packages are built against the LibYAML stable branch.
- Added attributes yaml.__version__ and yaml.__with_libyaml__ (fixed #85).

3.05 (2007-05-13)

- Allow for immutable subclasses of YAMLObject. Fixed #53.
- Make the encoding of the unicode->str conversion explicit; fixed #52.
- Fixed a problem when the DOCUMENT-END event is not emitted until the beginning of the next

document is available. Fixed [#51](#). Thanks [edward\(at\)sweetbytes.net](mailto:edward(at)sweetbytes.net) for the bug report.

- Improve output of float values. Fixed [#49](#).
- Fix the bug when the path in `add_path_resolver` contains boolean values. Fixed [#43](#) (thanks to [jstroud\(at\)mbi.ucla.edu](mailto:jstroud(at)mbi.ucla.edu) for reporting and pointing to the cause).
- Use the `types` module instead of constructing type objects by hand. Fixed [#41](#). Thanks to [v.haisman\(at\)sh.cvut.cz](mailto:v.haisman(at)sh.cvut.cz) for the patch.
- Fix loss of microsecond precision in `datetime.datetime` constructor (fix [#30](#)). Thanks to [edemaine\(at\)mit.edu](mailto:edemaine(at)mit.edu) for the bug report and the patch.
- Fixed loading an empty YAML stream.

3.04 (2006-08-20)

- Include experimental [LibYAML](#) bindings.
- Fully support recursive structures (close [#5](#)).
- Sort dictionary keys (close [#23](#)). Mapping node values are now represented as lists of pairs instead of dictionaries. Do not check for duplicate mapping keys as it didn't work correctly anyway.
- Fix invalid output of single-quoted scalars in cases when a single quote is not escaped when preceded by whitespaces or line breaks (close [#17](#)).
- To make porting easier, rewrite Parser not using generators.
- Fix handling of unexpected block mapping values, like

```
: foo
```

- Fix a bug in `Representer.represent_object`: `copy_reg.dispatch_table` was not correctly handled.
- Fix a bug when a block scalar is incorrectly emitted in the simple key context.
- Hold references to the objects being represented (close [#22](#)).
- Make `Representer` not try to guess `!!pairs` when a list is represented.
- Fix timestamp constructing and representing (close [#25](#)).
- Fix the 'N' plain scalar being incorrectly recognized as `!!bool` (close [#26](#)).

3.03 (2006-06-19)

- Fix Python 2.5 compatibility issues.
- Fix numerous bugs in the float handling.
- Fix scanning some ill-formed documents.
- Other minor fixes.

3.02 (2006-05-15)

- Fix win32 installer. Apparently `bdist_wininst` does not work well under Linux.
- Fix a bug in `add_path_resolver`.
- Add the `yaml-highlight` example. Try to run on a color terminal: `python yaml_h1.py <any_document.yaml`.

3.01 (2006-05-07)

- Initial release. The version number reflects the codename of the project (PyYAML 3000) and differentiates it from the abandoned [PyYaml](#) module.

Development and bug reports

You may check out the PyYAML source code from [PyYAML SVN repository](#).

If you find a bug in PyYAML, please [file a bug report](#). You may review open bugs through [the list of open tickets](#).

You may discuss PyYAML at [the YAML-core mailing list](#).

Author and copyright

The PyYAML module is written by Kirill Simonov.

PyYAML is released under the MIT license.