
pureyaml Documentation

Release 0.1.0

Manu Phatak

January 24, 2016

1	Contents:	1
1.1	pureyaml	1
1.1.1	Features	1
1.1.2	Credits	1
1.2	Installation	1
1.3	Usage	2
1.4	Contributing	2
1.4.1	Types of Contributions	2
	Report Bugs	2
	Fix Bugs	3
	Implement Features	3
	Write Documentation	3
	Submit Feedback	3
1.4.2	Get Started!	3
1.4.3	Pull Request Guidelines	4
1.4.4	Tips	4
1.5	Credits	4
1.5.1	Development Lead	4
1.5.2	Contributors	4
1.6	History	4
1.6.1	Next Release	4
1.6.2	0.1.0 (2016-01-xx)	4
1.7	pureyaml package	5
1.7.1	pureyaml	5
1.7.2	Subpackages	5
	pureyaml._compat package	5
	pureyaml.grammar package	16
	pureyaml.ply package	20
1.7.3	Submodules	29
1.7.4	pureyaml.decoder module	29
1.7.5	pureyaml.encoder module	30
1.7.6	pureyaml.exceptions module	31
1.7.7	pureyaml.nodes module	32
1.7.8	pureyaml.parser module	36
2	Feedback	37
3	Indices and tables	39

Contents:

1.1 pureyaml

Yet another yaml parser, in pure python.

1.1.1 Features

- Documentation: <https://pureyaml.readthedocs.org>
- Open Source: <https://github.com/bionikspoon/pureyaml>
- MIT license
- YAML encoder/decoder written in pure python

1.1.2 Credits

Tools used in rendering this package:

- Cookiecutter
- [bionikspoon/cookiecutter-pypackage](#) forked from [audreyr/cookiecutter-pypackage](#)

1.2 Installation

At the command line either via easy_install or pip:

```
$ pip install pureyaml
```

```
$ easy_install pureyaml
```

Or, if you have virtualenvwrapper installed:

```
$ mkvirtualenv pureyaml
$ pip install pureyaml
```

Uninstall:

```
$ pip uninstall pureyaml
```

1.3 Usage

To use pureyaml in a project:

```
import pureyaml

>>> import pureyaml
>>> from textwrap import dedent
>>> from pprint import pprint
>>> text = dedent("""
...     marvel:
...     - iron man
...     - the hulk
...     - captain america
...     dc:
...     - batman
...     - the joker
...     - superman
... """)[1:]

>>> pprint(pureyaml.load(text))
{'dc': ['batman', 'the joker', 'superman'],
 'marvel': ['iron man', 'the hulk', 'captain america']}

>>> print(pureyaml.dump(pureyaml.load(text)))
dc:
- batman
- the joker
- superman
marvel:
- iron man
- the hulk
- captain america
```

1.4 Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

1.4.1 Types of Contributions

Report Bugs

Report bugs at <https://github.com/bionikspoon/pureyaml/issues>.

If you are reporting a bug, please include:

- Your operating system name and version.
- Any details about your local setup that might be helpful in troubleshooting.
- Detailed steps to reproduce the bug.

Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with “bug” is open to whoever wants to implement it.

Implement Features

Look through the GitHub issues for features. Anything tagged with “feature” is open to whoever wants to implement it.

Write Documentation

pureyaml could always use more documentation, whether as part of the official pureyaml docs, in docstrings, or even on the web in blog posts, articles, and such.

Submit Feedback

The best way to send feedback is to file an issue at <https://github.com/bionikspoon/pureyaml/issues>.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

1.4.2 Get Started!

Ready to contribute? Here’s how to set up *pureyaml* for local development.

1. Fork the *pureyaml* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/pureyaml.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv pureyaml
$ cd pureyaml/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b feature/name-of-your-feature
$ git checkout -b hotfix/name-of-your-bugfix
```

Now you can make your changes locally.

5. When you’re done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 pureyaml tests
$ python setup.py test
$ tox
```

To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

1.4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests.
2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.6, 2.7, 3.3, 3.4, 3.5, and PyPy. Check https://travis-ci.org/bionikspoon/pureyaml/pull_requests and make sure that the tests pass for all supported Python versions.

1.4.4 Tips

To run a subset of tests:

```
$ py.test tests/test_pureyaml.py
```

1.5 Credits

1.5.1 Development Lead

- Manu Phatak <bionikspoon@gmail.com>

1.5.2 Contributors

None yet. Why not be the first?

1.6 History

1.6.1 Next Release

- Coming Soon

1.6.2 0.1.0 (2016-01-xx)

- Coming Soon

1.7 pureyaml package

1.7.1 pureyaml

Yet another yaml parser, in pure python.

`pureyaml.dump(obj, fp=None, indent=None, sort_keys=False, **kw)`
 Dump object to a file like object or string.

Parameters

- **obj** –
- **fp** – Open file like object
- **indent** (*int*) – Indent size, default 2
- **sort_keys** (*bool*) – Optionally sort dictionary keys.

Returns Yaml serialized data.

`pureyaml.dumps(obj, indent=None, default=None, sort_keys=False, **kw)`
 Dump string.

`pureyaml.load(s, **kwargs)`
 Load yaml file

`pureyaml.loads(s, cls=None, **kwargs)`
 Load string

1.7.2 Subpackages

pureyaml._compat package

Python 2to3 compatibility handling.

`class pureyaml._compat.NullHandler(level=0)`
 Bases: `logging.Handler`

This handler does nothing. It's intended to be used to avoid the “No handlers could be found for logger XXX” one-off warning. This is important for library code, which may contain code to log events. If a user of the library does not configure logging, the one-off warning might be produced; to avoid this, the library developer simply needs to instantiate a NullHandler and add it to the top-level logger of the library module or package.

`__doc__` = ‘\n This handler does nothing. It’s intended to be used to avoid the\n “No handlers could be found for logger

`__module__` = ‘logging’

`createLock()`

`emit(record)`

`handle(record)`

`class pureyaml._compat.collections_abc`

`class Callable`

Bases: `object`

`__abstractmethods__` = `frozenset(['__call__'])`

```
__call__ (*args, **kwargs)

__dict__ = dict_proxy({'_abc_cache': <_weakrefset.WeakSet object at 0x7fb588dcf6d0>, '__module__': '_abcoll',
__doc__ = None

__metaclass__
    alias of ABCMeta

__module__ = '_abcoll'
classmethod __subclasshook__ (C)

__weakref__
    list of weak references to the object (if defined)

__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 1
__abc_registry = <_weakrefset.WeakSet object>

class collections_abc.Container
    Bases: object

    __abstractmethods__ = frozenset(['__contains__'])

    __contains__ (x)

    __dict__ = dict_proxy({'_abc_cache': <_weakrefset.WeakSet object at 0x7fb588dcf510>, '__module__': '_abcoll',
__doc__ = None

__metaclass__
    alias of ABCMeta

__module__ = '_abcoll'
classmethod __subclasshook__ (C)

__weakref__
    list of weak references to the object (if defined)

__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 1
__abc_registry = <_weakrefset.WeakSet object>

class collections_abc.Hashable
    Bases: object

    __abstractmethods__ = frozenset(['__hash__'])

    __dict__ = dict_proxy({'_abc_cache': <_weakrefset.WeakSet object at 0x7fb588dc9dd0>, '__module__': '_abcoll',
__doc__ = None

    __hash__ ()

    __metaclass__
        alias of ABCMeta

    __module__ = '_abcoll'
```

```

classmethod __subclasshook__(C)

__weakref__
    list of weak references to the object (if defined)

__abc_cache = <_weakrefset.WeakSet object>

__abc_negative_cache = <_weakrefset.WeakSet object>

__abc_negative_cache_version = 0

__abc_registry = <_weakrefset.WeakSet object>

class collections_abc.ItemsView(mapping)
    Bases: _abcoll.MappingView, _abcoll.Set

    __abstractmethods__ = frozenset([])

    __contains__(item)

    __doc__ = None

    __iter__()

    __module__ = '_abcoll'

    __abc_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache_version = 3

    __abc_registry = <_weakrefset.WeakSet object>

    classmethod __from_iterable(it)

class collections_abc.Iterable
    Bases: object

    __abstractmethods__ = frozenset(['__iter__'])

    __dict__ = dict_proxy({'__abc_cache': <_weakrefset.WeakSet object at 0x7fb588dc9f90>, '__module__': '_abcoll',
    __doc__ = None

    __iter__()

    __metaclass__
        alias of ABCMeta

    __module__ = '_abcoll'

    classmethod __subclasshook__(C)

    __weakref__
        list of weak references to the object (if defined)

    __abc_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache_version = 0

    __abc_registry = <_weakrefset.WeakSet object>

class collections_abc.Iterator
    Bases: _abcoll.Iterable

    __abstractmethods__ = frozenset(['next'])

```

```
__doc__ = None
__iter__ ()
__module__ = '_abcoll'
classmethod __subclasshook__ (C)
__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 1
__abc_registry = <_weakrefset.WeakSet object>
next ()
    Return the next item from the iterator. When exhausted, raise StopIteration
class collections_abc.KeysView (mapping)
    Bases: _abcoll.MappingView, _abcoll.Set
    __abstractmethods__ = frozenset([])
    __contains__ (key)
    __doc__ = None
    __iter__ ()
    __module__ = '_abcoll'
    __abc_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache_version = 3
    __abc_registry = <_weakrefset.WeakSet object>
    classmethod _from_iterable (it)
class collections_abc.Mapping
    Bases: _abcoll.Sized, _abcoll.Iterable, _abcoll.Container
    A Mapping is a generic container for associating key/value pairs.
    This class provides concrete generic implementations of all methods except for __getitem__, __iter__, and
    __len__.
    __abstractmethods__ = frozenset(['__iter__', '__getitem__', '__len__'])
    __contains__ (key)
    __doc__ = 'A Mapping is a generic container for associating key/value\n pairs.\n\n This class provides concrete gen
    __eq__ (other)
    __getitem__ (key)
    __hash__ = None
    __module__ = '_abcoll'
    __ne__ (other)
    __abc_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache = <_weakrefset.WeakSet object>
```

```

    _abc_negative_cache_version = 25
    _abc_registry = <_weakrefset.WeakSet object>
    get (k[, d]) → D[k] if k in D, else d. d defaults to None.
    items () → list of D's (key, value) pairs, as 2-tuples
    iteritems () → an iterator over the (key, value) items of D
    iterkeys () → an iterator over the keys of D
    itervalues () → an iterator over the values of D
    keys () → list of D's keys
    values () → list of D's values

class collections_abc.MappingView(mapping)
    Bases: _abcoll.Sized
    __abstractmethods__ = frozenset([])
    __doc__ = None
    __init__(mapping)
    __len__()
    __module__ = '_abcoll'
    __repr__()
    _abc_cache = <_weakrefset.WeakSet object>
    _abc_negative_cache = <_weakrefset.WeakSet object>
    _abc_negative_cache_version = 3
    _abc_registry = <_weakrefset.WeakSet object>

class collections_abc.MutableMapping
    Bases: _abcoll.Mapping
    A MutableMapping is a generic container for associating key/value pairs.
    This class provides concrete generic implementations of all methods except for __getitem__, __setitem__,
    __delitem__, __iter__, and __len__.
    _MutableMapping_marker = <object object>
    __abstractmethods__ = frozenset(['__delitem__', '__setitem__', '__getitem__', '__iter__', '__len__'])
    __delitem__(key)
    __doc__ = 'A MutableMapping is a generic container for associating\n key/value pairs.\n\n This class provides cono
    __module__ = '_abcoll'
    __setitem__(key, value)
    _abc_cache = <_weakrefset.WeakSet object>
    _abc_negative_cache = <_weakrefset.WeakSet object>
    _abc_negative_cache_version = 25
    _abc_registry = <_weakrefset.WeakSet object>
    clear () → None. Remove all items from D.

```

pop (*k*, *d*) → *v*, remove specified key and return the corresponding value.

If key is not found, *d* is returned if given, otherwise `KeyError` is raised.

popitem () → (*k*, *v*), remove and return some (key, value) pair

as a 2-tuple; but raise `KeyError` if *D* is empty.

setdefault (*k*, *d*) → *D.get(k,d)*, also set *D[k]=d* if *k* not in *D*

update ([*E*], ***F*) → `None`. Update *D* from mapping/iterable *E* and *F*.

If *E* present and has a `.keys()` method, does: for *k* in *E*: *D[k] = E[k]* If *E* present and lacks `.keys()` method, does: for (*k*, *v*) in *E*: *D[k] = v* In either case, this is followed by: for *k*, *v* in *F.items()*: *D[k] = v*

class `collections_abc.MutableSequence`

Bases: `_abcoll.Sequence`

All the operations on a read-only sequence.

Concrete subclasses must provide `__new__` or `__init__`, `__getitem__`, `__setitem__`, `__delitem__`, `__len__`, and `insert()`.

`__abstractmethods__` = `frozenset(['__delitem__', '__setitem__', '__getitem__', '__len__', 'insert'])`

`__delitem__` (*index*)

`__doc__` = 'All the operations on a read-only sequence.\n\n Concrete subclasses must provide __new__ or __init__,\n'

`__iadd__` (*values*)

`__module__` = `'_abcoll'`

`__setitem__` (*index*, *value*)

`__abc_cache` = `<_weakrefset.WeakSet object>`

`__abc_negative_cache` = `<_weakrefset.WeakSet object>`

`__abc_negative_cache_version` = 8

`__abc_registry` = `<_weakrefset.WeakSet object>`

append (*value*)

S.append(object) – append object to the end of the sequence

extend (*values*)

S.extend(iterable) – extend sequence by appending elements from the iterable

insert (*index*, *value*)

S.insert(index, object) – insert object before index

pop ([*index*]) → item – remove and return item at index (default last).

Raise `IndexError` if list is empty or index is out of range.

remove (*value*)

S.remove(value) – remove first occurrence of value. Raise `ValueError` if the value is not present.

reverse ()

S.reverse() – reverse *IN PLACE*

class `collections_abc.MutableSet`

Bases: `_abcoll.Set`

A mutable set is a finite, iterable container.

This class provides concrete generic implementations of all methods except for `__contains__`, `__iter__`, `__len__`, `add()`, and `discard()`.

To override the comparisons (presumably for speed, as the semantics are fixed), all you have to do is redefine `__le__` and then the other operations will automatically follow suit.

```

__abstractmethods__ = frozenset(['discard', 'add', '__iter__', '__len__', '__contains__'])

__doc__ = 'A mutable set is a finite, iterable container.\n\n This class provides concrete generic implementations of'

__iand__(it)

__ior__(it)

__isub__(it)

__ixor__(it)

__module__ = '_abcoll'

__abc_cache = <_weakrefset.WeakSet object>

__abc_negative_cache = <_weakrefset.WeakSet object>

__abc_negative_cache_version = 2

__abc_registry = <_weakrefset.WeakSet object>

add(value)
    Add an element.

clear()
    This is slow (creates N new iterators!) but effective.

discard(value)
    Remove an element. Do not raise an exception if absent.

pop()
    Return the popped value. Raise KeyError if empty.

remove(value)
    Remove an element. If not a member, raise a KeyError.

class collections_abc.Sequence
    Bases: _abcoll.Sized, _abcoll.Iterable, _abcoll.Container
    All the operations on a read-only sequence.
    Concrete subclasses must override __new__ or __init__, __getitem__, and __len__.

    __abstractmethods__ = frozenset(['__getitem__', '__len__'])

    __contains__(value)

    __doc__ = 'All the operations on a read-only sequence.\n\n Concrete subclasses must override __new__ or __init__'

    __getitem__(index)

    __iter__()

    __module__ = '_abcoll'

    __reversed__()

    __abc_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache = <_weakrefset.WeakSet object>

    __abc_negative_cache_version = 7

    __abc_registry = <_weakrefset.WeakSet object>

```

count (*value*) → integer – return number of occurrences of value

index (*value*) → integer – return first index of value.

Raises ValueError if the value is not present.

class collections_abc.**Set**

Bases: `_abcoll.Sized`, `_abcoll.Iterable`, `_abcoll.Container`

A set is a finite, iterable container.

This class provides concrete generic implementations of all methods except for `__contains__`, `__iter__` and `__len__`.

To override the comparisons (presumably for speed, as the semantics are fixed), all you have to do is redefine `__le__` and then the other operations will automatically follow suit.

`__abstractmethods__` = frozenset(['`__iter__`', '`__len__`', '`__contains__`'])

`__and__` (*other*)

`__doc__` = 'A set is a finite, iterable container.\n\n This class provides concrete generic implementations of all\n met'

`__eq__` (*other*)

`__ge__` (*other*)

`__gt__` (*other*)

`__hash__` = None

`__le__` (*other*)

`__lt__` (*other*)

`__module__` = '`_abcoll`'

`__ne__` (*other*)

`__or__` (*other*)

`__sub__` (*other*)

`__xor__` (*other*)

`__abc_cache` = <_weakrefset.WeakSet object>

`__abc_negative_cache` = <_weakrefset.WeakSet object>

`__abc_negative_cache_version` = 1

`__abc_registry` = <_weakrefset.WeakSet object>

classmethod `__from_iterable` (*it*)

Construct an instance of the class from any iterable input.

Must override this method if the class constructor signature does not accept an iterable for an input.

`__hash__` ()

Compute the hash value of a set.

Note that we don't define `__hash__`: not all sets are hashable. But if you define a hashable set type, its `__hash__` should call this function.

This must be compatible `__eq__`.

All sets ought to compare equal if they contain the same elements, regardless of how they are implemented, and regardless of the order of the elements; so there's not much freedom for `__eq__` or `__hash__`. We match the algorithm used by the built-in frozenset type.


```

isdisjoint (other)
    Return True if two sets have a null intersection.

class collections_abc.Sized
    Bases: object

    __abstractmethods__ = frozenset(['__len__'])
    __dict__ = dict_proxy({'__abc_cache': <_weakrefset.WeakSet object at 0x7fb588dcf350>, '__module__': '_abcoll',
    __doc__ = None
    __len__ ()
    __metaclass__
        alias of ABCMeta
    __module__ = '_abcoll'
    classmethod __subclasshook__ (C)
    __weakref__
        list of weak references to the object (if defined)
    __abc_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache_version = 1
    __abc_registry = <_weakrefset.WeakSet object>

class collections_abc.ValuesView (mapping)
    Bases: _abcoll.MappingView

    __abstractmethods__ = frozenset([])
    __contains__ (value)
    __doc__ = None
    __iter__ ()
    __module__ = '_abcoll'
    __abc_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache = <_weakrefset.WeakSet object>
    __abc_negative_cache_version = 3
    __abc_registry = <_weakrefset.WeakSet object>

collections_abc.__doc__ = None
collections_abc.__module__ = 'pureyaml._compat'

pureyaml._compat.total_ordering (cls)
    Class decorator that fills in missing ordering methods

pureyaml._compat.single_dispatch (function)
    Single-dispatch generic function decorator.

    Transforms a function into a generic function, which can have different behaviours depending upon the type of
    its first argument. The decorated function acts as the default implementation, and additional implementations
    can be registered using the register() attribute of the generic function.

```

Submodules

`pureyaml._compat.singledispatch` module

`pureyaml._compat.singledispatch.singledispatch` (*function*)

Single-dispatch generic function decorator.

Transforms a function into a generic function, which can have different behaviours depending upon the type of its first argument. The decorated function acts as the default implementation, and additional implementations can be registered using the `register()` attribute of the generic function.

`pureyaml._compat.singledispatch_helpers` module

class `pureyaml._compat.singledispatch_helpers.ChainMap` (**maps*)

Bases: `_abcoll.MutableMapping`

A ChainMap groups multiple dicts (or other mappings) together to create a single, updateable view.

The underlying mappings are stored in a list. That list is public and can accessed or updated using the *maps* attribute. There is no other state.

Lookups search the underlying mappings successively until a key is found. In contrast, writes, updates, and deletions only operate on the first mapping.

__abstractmethods__ = `frozenset([])`

__contains__ (*key*)

__copy__ ()

New ChainMap or subclass with a new copy of *maps*[0] and refs to *maps*[1:]

__delitem__ (*key*)

__doc__ = `u'A ChainMap groups multiple dicts (or other mappings) together\n to create a single, updateable view.\n\n'`

__getitem__ (*key*)

__init__ (**maps*)

Initialize a ChainMap by setting *maps* to the given mappings. If no mappings are provided, a single empty dictionary is used.

__iter__ ()

__len__ ()

__missing__ (*key*)

__module__ = `'pureyaml._compat.singledispatch_helpers'`

__repr__ ()

__setitem__ (*key, value*)

_abc_cache = `<_weakrefset.WeakSet object>`

_abc_negative_cache = `<_weakrefset.WeakSet object>`

_abc_negative_cache_version = `25`

_abc_registry = `<_weakrefset.WeakSet object>`

clear ()

Clear *maps*[0], leaving *maps*[1:] intact.

```

copy ()
    New ChainMap or subclass with a new copy of maps[0] and refs to maps[1:]

classmethod fromkeys (iterable, *args)
    Create a ChainMap with a single dict created from the iterable.

get (key, default=None)

new_child ()
    New ChainMap with a new dict followed by all previous maps.

parents
    New ChainMap from maps[1:].

pop (key, *args)
    Remove key from maps[0] and return its value. Raise KeyError if key not in maps[0].

popitem ()
    Remove and return an item pair from maps[0]. Raise KeyError if maps[0] is empty.

class pureyaml._compat.singledispatch_helpers.MappingProxyType (data)
    Bases: UserDict.UserDict

    __doc__ = None

    __init__ (data)

    __module__ = 'pureyaml._compat.singledispatch_helpers'

class pureyaml._compat.singledispatch_helpers.Support
    Bases: object

    __dict__ = dict_proxy({'dummy': <function dummy at 0x7fb585a13de8>, '__module__': 'pureyaml._compat.singledispatch_helpers'})

    __doc__ = None

    __module__ = 'pureyaml._compat.singledispatch_helpers'

    __weakref__
        list of weak references to the object (if defined)

    cpython_only (func)

    dummy ()

pureyaml._compat.singledispatch_helpers.get_cache_token ()

pureyaml._compat.singledispatch_helpers.recursive_repr (fillvalue=u'...')
    Decorator to make a repr function return fillvalue for a recursive call

```

pureyaml._compat.total_ordering module

Python2.7 functools.partial included for Python2.6

```

pureyaml._compat.total_ordering._ge_from_gt (self, other)
    Return a >= b. Computed by @total_ordering from (a > b) or (a == b).

pureyaml._compat.total_ordering._ge_from_le (self, other)
    Return a >= b. Computed by @total_ordering from (not a <= b) or (a == b).

pureyaml._compat.total_ordering._ge_from_lt (self, other)
    Return a >= b. Computed by @total_ordering from (not a < b).

```

`pureyaml._compat.total_ordering._gt_from_ge(self, other)`
Return `a > b`. Computed by `@total_ordering` from `(a >= b)` and `(a != b)`.

`pureyaml._compat.total_ordering._gt_from_le(self, other)`
Return `a > b`. Computed by `@total_ordering` from `(not a <= b)`.

`pureyaml._compat.total_ordering._gt_from_lt(self, other)`
Return `a > b`. Computed by `@total_ordering` from `(not a < b)` and `(a != b)`.

`pureyaml._compat.total_ordering._le_from_ge(self, other)`
Return `a <= b`. Computed by `@total_ordering` from `(not a >= b)` or `(a == b)`.

`pureyaml._compat.total_ordering._le_from_gt(self, other)`
Return `a <= b`. Computed by `@total_ordering` from `(not a > b)`.

`pureyaml._compat.total_ordering._le_from_lt(self, other)`
Return `a <= b`. Computed by `@total_ordering` from `(a < b)` or `(a == b)`.

`pureyaml._compat.total_ordering._lt_from_ge(self, other)`
Return `a < b`. Computed by `@total_ordering` from `(not a >= b)`.

`pureyaml._compat.total_ordering._lt_from_gt(self, other)`
Return `a < b`. Computed by `@total_ordering` from `(not a > b)` and `(a != b)`.

`pureyaml._compat.total_ordering._lt_from_le(self, other)`
Return `a < b`. Computed by `@total_ordering` from `(a <= b)` and `(a != b)`.

`pureyaml._compat.total_ordering.total_ordering(cls)`
Class decorator that fills in missing ordering methods

pureyaml.grammar package

Submodules

pureyaml.grammar._parsetab module

pureyaml.grammar.productions module

Yaml grammar production rules.

```
class pureyaml.grammar.productions.YAMLProductions
    Bases: pureyaml.grammar.tokens.YAMLTokens
    __doc__ = None
    __module__ = 'pureyaml.grammar.productions'
    p_collection(p)
        collection [sequence]
            map
            flow_collection
    p_doc(p)
        doc [collection]
        scalar
    p_doc__indent(p)
```

```

    doc [DOC_START doc DOC_END]
        DOC_START doc
        INDENT doc DEDENT

p_doc_scalar_collection_ignore (p)
    scalar : ignore_indent_dedent scalar

p_docs__init (p)
    docs : docs doc

p_docs__last (p)
    docs [doc]
        doc DOC_END

p_flow_collection (p)
    flow_collection [F_SEQUENCE_START flow_sequence F_SEQUENCE_END]
        F_SEQUENCE_START flow_sequence F_SEP F_SEQUENCE_END
        F_MAP_START flow_map F_MAP_END
        F_MAP_START flow_map F_SEP F_MAP_END

p_flow_map__init (p)
    flow_map : flow_map F_SEP flow_map_item

p_flow_map__last (p)
    flow_map : flow_map_item

p_flow_map_item (p)
    flow_map_item : flow_map_item_key flow_map_item_value

p_flow_map_item_key (p)
    flow_map_item_key : scalar F_MAP_KEY

p_flow_map_item_value (p)
    flow_map_item_value : scalar

p_flow_sequence__init (p)
    flow_sequence : flow_sequence F_SEP flow_sequence_item

p_flow_sequence__last (p)
    flow_sequence : flow_sequence_item

p_flow_sequence_item (p)
    flow_sequence_item : scalar

p_ignore_indent_dedent (p)
    ignore_indent_dedent : INDENT DEDENT

p_map__init (p)
    map : map map_item

p_map__last (p)
    map : map_item

p_map_item (p)
    map_item : map_item_key map_item_value

p_map_item__key_value_collection (p)
    map_item_key : B_MAP_KEY INDENT collection DEDENT map_item_value : B_MAP_VALUE IN-
    DENT collection DEDENT

```

p_map_item_compact_scalar (*p*)
map_item : B_MAP_COMPACT_KEY scalar B_MAP_VALUE scalar DEDENT

p_map_item_key_map_item_value_sequence_item_compact_collection (*p*)
map_item_key : B_MAP_COMPACT_KEY collection DEDENT map_item_value
: B_MAP_COMPACT_VALUE collection DEDENT sequence_item :
B_SEQUENCE_COMPACT_START collection DEDENT

p_map_item_key (*p*)
map_item_key : scalar

p_map_item_key_complex_key_scalar (*p*)
map_item_key : B_MAP_KEY scalar

p_map_item_value_flow_collection (*p*)
map_item_value : B_MAP_VALUE flow_collection

p_map_item_value_scalar (*p*)
map_item_value : B_MAP_VALUE scalar

p_map_item_value_scalar_indented (*p*)
map_item_value : B_MAP_VALUE INDENT scalar DEDENT

p_map_item_value_sequence_no_indent (*p*)
map_item_value : B_MAP_VALUE sequence

p_scalar (*p*)
scalar : SCALAR

p_scalar_doublequote (*p*)
scalar : DOUBLEQUOTE_START SCALAR DOUBLEQUOTE_END

p_scalar_explicit_cast (*p*)
scalar : CAST_TYPE scalar

p_scalar_folded (*p*)
scalar : B_FOLD_START scalar_group B_FOLD_END

p_scalar_indented_flow (*p*)
scalar : INDENT scalar_group DEDENT

p_scalar_literal (*p*)
scalar : B_LITERAL_START scalar_group B_LITERAL_END

p_scalar_quote_empty (*p*)
scalar [DOUBLEQUOTE_START DOUBLEQUOTE_END]
SINGLEQUOTE_START SINGLEQUOTE_END

p_scalar_singlequote (*p*)
scalar : SINGLEQUOTE_START SCALAR SINGLEQUOTE_END

p_scalar_string_indented_multi_line (*p*)
scalar : scalar INDENT SCALAR DEDENT

p_scalar_group (*p*)
scalar_group [SCALAR]
scalar_group SCALAR

p_sequence_init (*p*)
sequence : sequence sequence_item

```

p_sequence__last (p)
    sequence : sequence_item

p_sequence_item_collection (p)
    sequence_item : B_SEQUENCE_START INDENT collection DEDENT

p_sequence_item_flow_collection (p)
    sequence_item : B_SEQUENCE_START flow_collection

p_sequence_item_scalar (p)
    sequence_item : B_SEQUENCE_START scalar

```

`pureyaml.grammar.tokens` module

Yaml tokens.

```

class pureyaml.grammar.tokens.TokenList
    Bases: object

    __dict__ = dict_proxy({'tokens': ['DOC_START', 'DOC_END', 'B_SEQUENCE_COMPACT_START', 'B_SEQUENCE_START', 'B_MAP_START', 'B_MAP_END', 'B_SEQUENCE_END', 'B_MAP_END']})
    __doc__ = None
    __module__ = 'pureyaml.grammar.tokens'
    __weakref__ = list of weak references to the object (if defined)

    tokens = ['DOC_START', 'DOC_END', 'B_SEQUENCE_COMPACT_START', 'B_SEQUENCE_START', 'B_MAP_START', 'B_MAP_END', 'B_SEQUENCE_END', 'B_MAP_END']

```

`pureyaml.grammar.utils` module

```

pureyaml.grammar.utils.find_column (t)
    Get cursor position, based on previous newline

class pureyaml.grammar.utils.fold
    Bases: object

    Properly clean fold text.

    __dict__ = dict_proxy({'__module__': 'pureyaml.grammar.utils', '__new__': <staticmethod object at 0x7fb585a357f8>})
    __doc__ = 'Properly clean “fold“ text.'
    __module__ = 'pureyaml.grammar.utils'
    static __new__ (text)
    __weakref__ = list of weak references to the object (if defined)

    classmethod _fold_replace (match)
    _re_fold_replace = <_sre.SRE_Pattern object>

pureyaml.grammar.utils.rollback_lexpos (t)

pureyaml.grammar.utils.strict (*types)
    Decorator, type check production rule output

```

pureyaml.ply package

Submodules

pureyaml.ply.cpp module

`pureyaml.ply.cpp.CPP_INTEGER(t)`

`(((((0x)(0X))[0-9a-fA-F]+)(d+))([uU][lL]|[lL][uU]|[uU][lL])?)`

`class pureyaml.ply.cpp.Macro(name, value, arglist=None, variadic=False)`

Bases: `object`

`__dict__ = dict_proxy({'__dict__': <attribute '__dict__' of 'Macro' objects>, '__module__': 'pureyaml.ply.cpp', '__we`

`__doc__ = None`

`__init__(name, value, arglist=None, variadic=False)`

`__module__ = 'pureyaml.ply.cpp'`

`__weakref__`

list of weak references to the object (if defined)

`class pureyaml.ply.cpp.Preprocessor(lexer=None)`

Bases: `object`

`__dict__ = dict_proxy({'__module__': 'pureyaml.ply.cpp', 'add_path': <function add_path at 0x7fb582df1578>, 'error`

`__doc__ = None`

`__init__(lexer=None)`

`__module__ = 'pureyaml.ply.cpp'`

`__weakref__`

list of weak references to the object (if defined)

`add_path(path)`

`collect_args(tokenlist)`

`define(tokens)`

`error(file, line, msg)`

`evalexpr(tokens)`

`expand_macros(tokens, expanded=None)`

`group_lines(input)`

`include(tokens)`

`lexprobe()`

`macro_expand_args(macro, args)`

`macro_prescan(macro)`

`parse(input, source=None, ignore={})`

`parsegen(input, source=None)`

`token()`

`tokenize(text)`


```

tokenstrip (tokens)
undef (tokens)
pureyaml.ply.cpp.t_CPP_CHAR (t)
    (L)?'([^\n]|(\.ln))*?'
pureyaml.ply.cpp.t_CPP_COMMENT1 (t)
    (*(ln)*?*/)
pureyaml.ply.cpp.t_CPP_COMMENT2 (t)
    (//.*?(nl$))
pureyaml.ply.cpp.t_CPP_INTEGER (t)
    (((((0x)|(0X))[0-9a-fA-F]+)|(d+))([uU][iL]|[iL][uU]|[uU][iL])?)
pureyaml.ply.cpp.t_CPP_STRING (t)
    "([^\n]|(\.ln))*?"
pureyaml.ply.cpp.t_CPP_WS (t)
    s+
pureyaml.ply.cpp.t_error (t)
pureyaml.ply.cpp.trigraph (input)

```

pureyaml.ply.ctokens module

```

pureyaml.ply.ctokens.t_COMMENT (t)
    /*(ln)*?*/
pureyaml.ply.ctokens.t_CPPCOMMENT (t)
    //.*n

```

pureyaml.ply.lex module

```

exception pureyaml.ply.lex.LexError (message, s)
    Bases: exceptions.Exception

```

```

    __doc__ = None
    __init__ (message, s)
    __module__ = 'pureyaml.ply.lex'
    __weakref__
        list of weak references to the object (if defined)

```

```

class pureyaml.ply.lex.LexToken
    Bases: object

```

```

    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.lex', '__str__': <function __str__ at 0x7fb5859c37d0>, '__repr__'
    __doc__ = None
    __module__ = 'pureyaml.ply.lex'
    __repr__ ()
    __str__ ()
    __weakref__
        list of weak references to the object (if defined)

```

```
class pureyaml.ply.lex.Lexer
```

```
    __doc__ = None
    __init__ ()
    __iter__ ()
    __module__ = 'pureyaml.ply.lex'
    __next__ ()
    begin (state)
    clone (object=None)
    current_state ()
    input (s)
    next ()
    pop_state ()
    push_state (state)
    readtab (tabfile, fdict)
    skip (n)
    token ()
    writetab (lextab, outputdir='')
```

```
class pureyaml.ply.lex.LexerReflect (ldict, log=None, reflags=0)
```

```
    Bases: object
```

```
    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.lex', 'get_states': <function get_states at 0x7fb5859d2848>, 'valid
    __doc__ = None
    __init__ (ldict, log=None, reflags=0)
    __module__ = 'pureyaml.ply.lex'
    __weakref__
        list of weak references to the object (if defined)
    get_all ()
    get_literals ()
    get_rules ()
    get_states ()
    get_tokens ()
    validate_all ()
    validate_literals ()
    validate_module (module)
    validate_rules ()
    validate_tokens ()
```

```
class pureyaml.ply.lex.NullLogger
```

```
    Bases: object
```

```
    __call__ (*args, **kwargs)
```

```
    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.lex', '__getattr__': <function __getattr__ at 0x7fb5859c>})
```

```
    __doc__ = None
```

```
    __getattr__ (name)
```

```
    __module__ = 'pureyaml.ply.lex'
```

```
    __weakref__
```

```
        list of weak references to the object (if defined)
```

```
class pureyaml.ply.lex.PlyLogger (f)
```

```
    Bases: object
```

```
    __dict__ = dict_proxy({'info': <function critical at 0x7fb5859c3938>, '__dict__': <attribute '__dict__' of 'PlyLogger' object>})
```

```
    __doc__ = None
```

```
    __init__ (f)
```

```
    __module__ = 'pureyaml.ply.lex'
```

```
    __weakref__
```

```
        list of weak references to the object (if defined)
```

```
    critical (msg, *args, **kwargs)
```

```
    debug (msg, *args, **kwargs)
```

```
    error (msg, *args, **kwargs)
```

```
    info (msg, *args, **kwargs)
```

```
    warning (msg, *args, **kwargs)
```

```
pureyaml.ply.lex.TOKEN (r)
```

```
pureyaml.ply.lex.Token (r)
```

```
pureyaml.ply.lex._form_master_re (relist, reflags, ldict, toknames)
```

```
pureyaml.ply.lex._funcs_to_names (funclist, namelist)
```

```
pureyaml.ply.lex._get_regex (func)
```

```
pureyaml.ply.lex._names_to_funcs (namelist, fdict)
```

```
pureyaml.ply.lex._statetoken (s, names)
```

```
pureyaml.ply.lex.get_caller_module_dict (levels)
```

```
pureyaml.ply.lex.lex (module=None, object=None, debug=False, optimize=False, lextab='lextab',  
                      reflags=0, nowarn=False, outputdir=None, debuglog=None, errorlog=None)
```

```
pureyaml.ply.lex.runmain (lexer=None, data=None)
```

pureyaml.ply.yacc module

```
class pureyaml.ply.yacc.Grammar (terminals)
```

```
    Bases: object
```

```
    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', 'unused_precedence': <function unused_precedence at 0x7fb5859c>})
```

```
__doc__ = None
__getitem__(index)
__init__(terminals)
__len__()
__module__ = 'pureyaml.ply.yacc'
__weakref__
    list of weak references to the object (if defined)
__first(beta)
add_production(prodname, syms, func=None, file='', line=0)
build_lritems()
compute_first()
compute_follow(start=None)
find_unreachable()
infinite_cycles()
set_precedence(term, assoc, level)
set_start(start=None)
undefined_symbols()
unused_precedence()
unused_rules()
unused_terminals()

exception pureyaml.ply.yacc.GrammarError
    Bases: pureyaml.ply.yacc.YaccError
    __doc__ = None
    __module__ = 'pureyaml.ply.yacc'

exception pureyaml.ply.yacc.LALRError
    Bases: pureyaml.ply.yacc.YaccError
    __doc__ = None
    __module__ = 'pureyaml.ply.yacc'

class pureyaml.ply.yacc.LRGeneratedTable(grammar, method='LALR', log=None)
    Bases: pureyaml.ply.yacc.LRTable
    __doc__ = None
    __init__(grammar, method='LALR', log=None)
    __module__ = 'pureyaml.ply.yacc'
    add_lalr_lookaheads(C)
    add_lookaheads(lookbacks, followset)
    compute_follow_sets(ntrans, readsets, inclsets)
    compute_lookback_includes(C, trans, nullable)
```

```

    compute_nullable_nonterminals ()
    compute_read_sets (C, ntrans, nullable)
    dr_relation (C, trans, nullable)
    find_nonterminal_transitions (C)
    lr0_closure (I)
    lr0_goto (I, x)
    lr0_items ()
    lr_parse_table ()
    pickle_table (filename, signature='')
    reads_relation (C, trans, empty)
    write_table (tabmodule, outputdir='', signature='')

class pureyaml.ply.yacc.LRItem (p, n)
    Bases: object

    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', '__str__': <function __str__ at 0x7fb5859f28c0>, '__repr__': <function __repr__ at 0x7fb5859f28c0>})
    __doc__ = None
    __init__ (p, n)
    __module__ = 'pureyaml.ply.yacc'
    __repr__ ()
    __str__ ()
    __weakref__
        list of weak references to the object (if defined)

class pureyaml.ply.yacc.LRParser (lrtab, errorf)

    __doc__ = None
    __init__ (lrtab, errorf)
    __module__ = 'pureyaml.ply.yacc'
    disable_defaulted_states ()
    errok ()
    parse (input=None, lexer=None, debug=False, tracking=False, tokenfunc=None)
    parsedebug (input=None, lexer=None, debug=False, tracking=False, tokenfunc=None)
    parseopt (input=None, lexer=None, debug=False, tracking=False, tokenfunc=None)
    parseopt_notrack (input=None, lexer=None, debug=False, tracking=False, tokenfunc=None)
    restart ()
    set_defaulted_states ()

class pureyaml.ply.yacc.LRTable
    Bases: object

    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', 'read_pickle': <function read_pickle at 0x7fb5859f3320>, 'write_pickle': <function write_pickle at 0x7fb5859f3320>})

```

```
__doc__ = None
__init__ ()
__module__ = 'pureyaml.ply.yacc'
__weakref__
    list of weak references to the object (if defined)
bind_callables (pdict)
read_pickle (filename)
read_table (module)
class pureyaml.ply.yacc.MiniProduction (str, name, len, func, file, line)
    Bases: object
    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', 'bind': <function bind at 0x7fb5859f27d0>, '__str__': <fun
    __doc__ = None
    __init__ (str, name, len, func, file, line)
    __module__ = 'pureyaml.ply.yacc'
    __repr__ ()
    __str__ ()
    __weakref__
        list of weak references to the object (if defined)
    bind (pdict)
class pureyaml.ply.yacc.NullLogger
    Bases: object
    __call__ (*args, **kwargs)
    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', '__getattr__': <function __getattr__ at 0x7fb5859f27d0>,
    __doc__ = None
    __getattr__ (name)
    __module__ = 'pureyaml.ply.yacc'
    __weakref__
        list of weak references to the object (if defined)
class pureyaml.ply.yacc.ParserReflect (pdict, log=None)
    Bases: object
    __dict__ = dict_proxy({'validate_precedence': <function validate_precedence at 0x7fb5859f4398>, 'validate_modules':
    __doc__ = None
    __init__ (pdict, log=None)
    __module__ = 'pureyaml.ply.yacc'
    __weakref__
        list of weak references to the object (if defined)
    get_all ()
    get_error_func ()
```

```
get_pfunctions()
get_precedence()
get_start()
get_tokens()
signature()
validate_all()
validate_error_func()
validate_modules()
validate_pfunctions()
validate_precedence()
validate_start()
validate_tokens()

class pureyaml.ply.yacc.PlyLogger(f)
    Bases: object

    __dict__ = dict_proxy({'info': <function debug at 0x7fb5859e32a8>, '__dict__': <attribute '__dict__' of 'PlyLogger' ob
    __doc__ = None
    __init__(f)
    __module__ = 'pureyaml.ply.yacc'
    __weakref__
        list of weak references to the object (if defined)
    critical(msg, *args, **kwargs)
    debug(msg, *args, **kwargs)
    error(msg, *args, **kwargs)
    info(msg, *args, **kwargs)
    warning(msg, *args, **kwargs)

class pureyaml.ply.yacc.Production(number, name, prod, precedence=('right', 0), func=None,
                                   file='', line=0)
    Bases: object

    __dict__ = dict_proxy({'__module__': 'pureyaml.ply.yacc', '__nonzero__': <function __nonzero__ at 0x7fb5859f2488>
    __doc__ = None
    __getitem__(index)
    __init__(number, name, prod, precedence=('right', 0), func=None, file='', line=0)
    __len__()
    __module__ = 'pureyaml.ply.yacc'
    __nonzero__()
    __repr__()
    __str__()
```

```
    __weakref__
        list of weak references to the object (if defined)

    bind(pdict)

    lr_item(n)

    reduced = 0

exception pureyaml.ply.yacc.VersionError
    Bases: pureyaml.ply.yacc.YaccError

    __doc__ = None

    __module__ = 'pureyaml.ply.yacc'

exception pureyaml.ply.yacc.YaccError
    Bases: exceptions.Exception

    __doc__ = None

    __module__ = 'pureyaml.ply.yacc'

    __weakref__
        list of weak references to the object (if defined)

class pureyaml.ply.yacc.YaccProduction(s, stack=None)

    __doc__ = None

    __getitem__(n)

    __getslice__(i, j)

    __init__(s, stack=None)

    __len__()

    __module__ = 'pureyaml.ply.yacc'

    __setitem__(n, v)

    error()

    lexpos(n)

    lexspan(n)

    lineno(n)

    linespan(n)

    set_lineno(n, lineno)

class pureyaml.ply.yacc.YaccSymbol

    __doc__ = None

    __module__ = 'pureyaml.ply.yacc'

    __repr__()

    __str__()

pureyaml.ply.yacc.call_errorfunc(errorfunc, token, parser)

pureyaml.ply.yacc.digraph(X, R, FP)
```



```
pureyaml.ply.yacc.errok()
pureyaml.ply.yacc.format_result(r)
pureyaml.ply.yacc.format_stack_entry(r)
pureyaml.ply.yacc.get_caller_module_dict(levels)
pureyaml.ply.yacc.parse_grammar(doc, file, line)
pureyaml.ply.yacc.restart()
pureyaml.ply.yacc.rightmost_terminal(symbols, terminals)
pureyaml.ply.yacc.token()
pureyaml.ply.yacc.traverse(x, N, stack, F, X, R, FP)
pureyaml.ply.yacc.yacc(method='LALR', debug=True, module=None, tabmodule='parsetab',
                        start=None, check_recursion=True, optimize=False, write_tables=True,
                        debugfile='parser.out', outputdir=None, debuglog=None, errorlog=None,
                        picklefile=None)
```

pureyaml.ply.ygen module

```
pureyaml.ply.ygen.filter_section(lines, tag)
pureyaml.ply.ygen.get_source_range(lines, tag)
pureyaml.ply.ygen.main()
```

1.7.3 Submodules

1.7.4 pureyaml.decoder module

```
class pureyaml.decoder.YAMLDecoder(*args, **kwargs)
    Bases: pureyaml.nodes.NodeVisitor
    Convert node tree into python object.
    __doc__ = 'Convert node tree into python object.'
    __module__ = 'pureyaml.decoder'
    decode(s)
    visit_Binary(node)
    visit_Bool(node)
    visit_Doc(node)
    visit_Docs(node)
    visit_Float(node)
    visit_Int(node)
    visit_Map(node)
    visit_Null(_)
    visit_Scalar(node)
```

```
visit_Sequence (node)
```

```
visit_Str (node)
```

1.7.5 pureyaml.encoder module

```
class pureyaml.encoder.SYMBOL (name)
```

```
    __doc__ = None
```

```
    __init__ (name)
```

```
    __module__ = 'pureyaml.encoder'
```

```
    __repr__ ()
```

```
class pureyaml.encoder.YAMLEncoder (indent=None, sort_keys=None, **kw)
```

```
    Bases: pureyaml.nodes.NodeVisitor
```

```
    Convert node tree into string.
```

```
    __doc__ = 'Convert node tree into string.'
```

```
    __init__ (indent=None, sort_keys=None, **kw)
```

```
    __module__ = 'pureyaml.encoder'
```

```
    _encode (obj)
```

```
    encode (obj)
```

```
    iter_map_items (node)
```

```
    iterencode (obj)
```

```
    stack = []
```

```
    visit_Binary (node)
```

```
    visit_Bool (node)
```

```
    visit_Float (node)
```

```
    visit_Int (node)
```

```
    visit_Map (node)
```

```
    visit_Null (node)
```

```
    visit_Scalar (node)
```

```
    visit_Sequence (node)
```

```
    visit_Str (node)
```

```
pureyaml.encoder._ (obj)
```

```
pureyaml.encoder.is_float (string)
```

```
pureyaml.encoder.node_encoder (*args, **kw)
```

```
    Convert python object to node tree.
```

1.7.6 pureyaml.exceptions module

exception pureyaml.exceptions.YAMLCastTypeError (*message=None, cast=None*)

Bases: exceptions.TypeError, *pureyaml.exceptions.YAMLErrorException*

__doc__ = None

__init__ (*message=None, cast=None*)

__module__ = 'pureyaml.exceptions'

__weakref__

list of weak references to the object (if defined)

exception pureyaml.exceptions.YAMLErrorException (*message*)

Bases: exceptions.Exception

Base exception for package

__doc__ = 'Base exception for package'

__init__ (*message*)

__module__ = 'pureyaml.exceptions'

__str__ ()

__weakref__

list of weak references to the object (if defined)

exception pureyaml.exceptions.YAMLStrictTypeError (*token, types, func*)

Bases: exceptions.TypeError, *pureyaml.exceptions.YAMLErrorException*

__doc__ = None

__init__ (*token, types, func*)

__module__ = 'pureyaml.exceptions'

__str__ ()

__weakref__

list of weak references to the object (if defined)

exception pureyaml.exceptions.YAMLSyntaxError (*p, value=None*)

Bases: exceptions.SyntaxException, *pureyaml.exceptions.YAMLErrorException*

__doc__ = None

__init__ (*p, value=None*)

__module__ = 'pureyaml.exceptions'

__str__ ()

__weakref__

list of weak references to the object (if defined)

msg_lines ()

exception pureyaml.exceptions.YAMLUnknownSyntaxError

Bases: exceptions.SyntaxException, *pureyaml.exceptions.YAMLErrorException*

Unexpected syntax error

__doc__ = 'Unexpected syntax error'

__module__ = 'pureyaml.exceptions'

__weakref__
list of weak references to the object (if defined)

1.7.7 pureyaml.nodes module

Node definitions for intermediary node tree.

class pureyaml.nodes.**Node** (*value*, ****kwargs**)

Bases: object

__dict__ = dict_proxy({'__ne__': <function __ne__ at 0x7fb585a6d848>, '__module__': 'pureyaml.nodes', '__ge__': <f

__doc__ = None

__eq__ (*other*)

__ge__ (*other*)

x.__ge__(y) <==> x>=y

__gt__ (*other*)

__init__ (*value*, ****kwargs**)

__le__ (*other*)

x.__le__(y) <==> x<=y

__lt__ (*other*)

x.__lt__(y) <==> x<y

__module__ = 'pureyaml.nodes'

__ne__ (*other*)

__repr__ ()

__weakref__

list of weak references to the object (if defined)

init_value (**values*, ****kwargs**)

repr_value (*value*)

class pureyaml.nodes.**Collection** (**values*, ****kwargs**)

Bases: pureyaml.nodes.SequenceMixin, *pureyaml.nodes.Node*

__abstractmethods__ = frozenset([])

__add__ (*other*)

__doc__ = None

__init__ (**values*, ****kwargs**)

__module__ = 'pureyaml.nodes'

_abc_cache = <_weakrefset.WeakSet object>

_abc_negative_cache = <_weakrefset.WeakSet object>

_abc_negative_cache_version = 25

_abc_registry = <_weakrefset.WeakSet object>

init_value (**value*, ****kwargs**)

class pureyaml.nodes.**Docs** (**values*, ****kwargs**)

Bases: *pureyaml.nodes.Collection*

```

__abstractmethods__ = frozenset([])
__doc__ = None
__module__ = 'pureyaml.nodes'
__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 25
__abc_registry = <_weakrefset.WeakSet object>
class pureyaml.nodes.Doc(*values, **kwargs)
Bases: pureyaml.nodes.Collection
__abstractmethods__ = frozenset([])
__doc__ = None
__module__ = 'pureyaml.nodes'
__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 25
__abc_registry = <_weakrefset.WeakSet object>
class pureyaml.nodes.Sequence(*values, **kwargs)
Bases: pureyaml.nodes.Collection
__abstractmethods__ = frozenset([])
__doc__ = None
__module__ = 'pureyaml.nodes'
__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 25
__abc_registry = <_weakrefset.WeakSet object>
class pureyaml.nodes.Map(*args, **kwargs)
Bases: pureyaml.nodes.MappingMixin, pureyaml.nodes.Collection
__abstractmethods__ = frozenset([])
__doc__ = None
__eq__ (other)
__module__ = 'pureyaml.nodes'
__abc_cache = <_weakrefset.WeakSet object>
__abc_negative_cache = <_weakrefset.WeakSet object>
__abc_negative_cache_version = 25
__abc_registry = <_weakrefset.WeakSet object>
init_value(*values, **kwargs)

```

```
class pureyaml.nodes.Scalar(value, *args, **kwargs)
    Bases: pureyaml.nodes.Node

    __doc__ = None
    __eq__(other)
    __gt__(other)
    __init__(value, *args, **kwargs)
    __len__()
    __module__ = 'pureyaml.nodes'
    init_value(value, *args, **kwargs)
    type = NotImplemented

class pureyaml.nodes.Null(value, *args, **kwargs)
    Bases: pureyaml.nodes.Scalar

    __doc__ = None
    __module__ = 'pureyaml.nodes'
    init_value(*values, **kwargs)
    type = None

class pureyaml.nodes.Str(value, *args, **kwargs)
    Bases: pureyaml.nodes.Scalar

    __doc__ = None
    __module__ = 'pureyaml.nodes'
    init_value(value, *args, **kwargs)
    type
        alias of str

class pureyaml.nodes.Int(value, *args, **kwargs)
    Bases: pureyaml.nodes.Scalar

    __doc__ = None
    __module__ = 'pureyaml.nodes'
    init_value(value, base=None, *args, **kwargs)
    type
        alias of int

class pureyaml.nodes.Float(value, *args, **kwargs)
    Bases: pureyaml.nodes.Scalar

    __doc__ = None
    __eq__(other)
    __module__ = 'pureyaml.nodes'
    init_value(value, *args, **kwargs)
    type
        alias of float
```

pureyaml Documentation, Release 0.1.0

1.7.8 pureyaml.parser module

Build yacc parser from lex tokens.

```
class pureyaml.parser.YAMLLexer
    Bases: pureyaml.grammar.tokens.YAMLTokens
    __doc__ = None
    __module__ = 'pureyaml.parser'
    classmethod build (**kwargs)
    t_ANY_error (t)
    classmethod tokenize (data)

class pureyaml.parser.YAMLParser (**kwargs)
    Bases: pureyaml.grammar productions.YAMLProductions
    __doc__ = None
    __init__ (**kwargs)
    __module__ = 'pureyaml.parser'
    p_error (p)
    parse (data, **kwargs)
    parsedebug (data, **kwargs)
    tokenize (data)
```

Feedback

If you have any suggestions or questions about **pureyaml** feel free to email me at bionikspoon@gmail.com.

If you encounter any errors or problems with **pureyaml**, please let me know! Open an Issue at the GitHub <https://github.com/bionikspoon/pureyaml> main repository.

Indices and tables

- `genindex`
- `modindex`
- `search`

p

- `pureyaml`, [5](#)
- `pureyaml._compat`, [5](#)
- `pureyaml._compat.singledispatch`, [14](#)
- `pureyaml._compat.singledispatch_helpers`,
[14](#)
- `pureyaml._compat.total_ordering`, [15](#)
- `pureyaml.decoder`, [29](#)
- `pureyaml.encoder`, [30](#)
- `pureyaml.exceptions`, [31](#)
- `pureyaml.grammar`, [16](#)
- `pureyaml.grammar._parsetab`, [16](#)
- `pureyaml.grammar.productions`, [16](#)
- `pureyaml.grammar.tokens`, [19](#)
- `pureyaml.grammar.utils`, [19](#)
- `pureyaml.nodes`, [32](#)
- `pureyaml.parser`, [36](#)
- `pureyaml.ply`, [20](#)
- `pureyaml.ply.cpp`, [20](#)
- `pureyaml.ply.ctokens`, [21](#)
- `pureyaml.ply.lex`, [21](#)
- `pureyaml.ply.yacc`, [23](#)
- `pureyaml.ply.ygen`, [29](#)

Symbols

- `__()` (in module `pureyaml.encoder`), 30
- `__MutableMapping__` marker
 - (`pureyaml._compat.collections_abc.MutableMapping` attribute), 9
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Callable` attribute), 5
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Container` attribute), 6
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Hashable` attribute), 6
- `__abstractmethods__` (`pureyaml._compat.collections_abc.ItemsView` attribute), 7
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Iterable` attribute), 7
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Iterator` attribute), 7
- `__abstractmethods__` (`pureyaml._compat.collections_abc.KeysView` attribute), 8
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Mapping` attribute), 8
- `__abstractmethods__` (`pureyaml._compat.collections_abc.MappingView` attribute), 9
- `__abstractmethods__` (`pureyaml._compat.collections_abc.MutableMapping` attribute), 9
- `__abstractmethods__` (`pureyaml._compat.collections_abc.MutableSequence` attribute), 10
- `__abstractmethods__` (`pureyaml._compat.collections_abc.MutableSet` attribute), 11
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Sequence` attribute), 11
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Set` attribute), 12
- `__abstractmethods__` (`pureyaml._compat.collections_abc.Sized` attribute), 13
- `__abstractmethods__` (`pureyaml._compat.collections_abc.ValuesView` attribute), 13
- `__abstractmethods__` (`pureyaml._compat singledispatch_helpers.ChainMap` attribute), 14
- `__abstractmethods__` (`pureyaml.nodes.Collection` attribute), 32
- `__abstractmethods__` (`pureyaml.nodes.Doc` attribute), 33
- `__abstractmethods__` (`pureyaml.nodes Docs` attribute), 32
- `__abstractmethods__` (`pureyaml.nodes.Map` attribute), 33
- `__abstractmethods__` (`pureyaml.nodes.Sequence` attribute), 33
- `__add__()` (`pureyaml.nodes.Collection` method), 32
- `__and__()` (`pureyaml._compat.collections_abc.Set` method), 12
- `__call__()` (`pureyaml._compat.collections_abc.Callable` method), 5
- `__call__()` (`pureyaml.ply.lex.NullLogger` method), 23
- `__call__()` (`pureyaml.ply.yacc.NullLogger` method), 26
- `__contains__()` (`pureyaml._compat.collections_abc.Container` method), 6
- `__contains__()` (`pureyaml._compat.collections_abc.ItemsView` method), 7
- `__contains__()` (`pureyaml._compat.collections_abc.KeysView` method), 8
- `__contains__()` (`pureyaml._compat.collections_abc.Mapping` method), 8
- `__contains__()` (`pureyaml._compat.collections_abc.Sequence` method), 11
- `__contains__()` (`pureyaml._compat.collections_abc.ValuesView` method), 13
- `__contains__()` (`pureyaml._compat singledispatch_helpers.ChainMap` method), 14
- `__contains__()` (`pureyaml._compat singledispatch_helpers.ChainMap` method), 14
- `__delitem__()` (`pureyaml._compat.collections_abc.MutableMapping` method), 9
- `__delitem__()` (`pureyaml._compat.collections_abc.MutableSequence` method), 10
- `__delitem__()` (`pureyaml._compat singledispatch_helpers.ChainMap` method), 14
- `__dict__` (`pureyaml._compat.collections_abc.Callable` attribute), 6
- `__dict__` (`pureyaml._compat.collections_abc.Container` attribute), 6
- `__dict__` (`pureyaml._compat.collections_abc.Hashable` attribute), 6

- `__dict__` (pureyaml._compat.collections_abc.Iterable attribute), 7
- `__dict__` (pureyaml._compat.collections_abc.Sized attribute), 13
- `__dict__` (pureyaml._compat.singledispatch_helpers.Support attribute), 15
- `__dict__` (pureyaml.grammar.tokens.TokenList attribute), 19
- `__dict__` (pureyaml.grammar.utils.fold attribute), 19
- `__dict__` (pureyaml.nodes.Node attribute), 32
- `__dict__` (pureyaml.nodes.NodeVisitor attribute), 35
- `__dict__` (pureyaml.nodes.ScalarDispatch attribute), 35
- `__dict__` (pureyaml.ply.cpp.Macro attribute), 20
- `__dict__` (pureyaml.ply.cpp.Preprocessor attribute), 20
- `__dict__` (pureyaml.ply.lex.LexToken attribute), 21
- `__dict__` (pureyaml.ply.lex.LexerReflect attribute), 22
- `__dict__` (pureyaml.ply.lex.NullLogger attribute), 23
- `__dict__` (pureyaml.ply.lex.PlyLogger attribute), 23
- `__dict__` (pureyaml.ply.yacc.Grammar attribute), 23
- `__dict__` (pureyaml.ply.yacc.LRItem attribute), 25
- `__dict__` (pureyaml.ply.yacc.LRTable attribute), 25
- `__dict__` (pureyaml.ply.yacc.MiniProduction attribute), 26
- `__dict__` (pureyaml.ply.yacc.NullLogger attribute), 26
- `__dict__` (pureyaml.ply.yacc.ParserReflect attribute), 26
- `__dict__` (pureyaml.ply.yacc.PlyLogger attribute), 27
- `__dict__` (pureyaml.ply.yacc.Production attribute), 27
- `__doc__` (pureyaml._compat.NullHandler attribute), 5
- `__doc__` (pureyaml._compat.collections_abc attribute), 13
- `__doc__` (pureyaml._compat.collections_abc.Callable attribute), 6
- `__doc__` (pureyaml._compat.collections_abc.Container attribute), 6
- `__doc__` (pureyaml._compat.collections_abc.Hashable attribute), 6
- `__doc__` (pureyaml._compat.collections_abc.ItemsView attribute), 7
- `__doc__` (pureyaml._compat.collections_abc.Iterable attribute), 7
- `__doc__` (pureyaml._compat.collections_abc.Iterator attribute), 7
- `__doc__` (pureyaml._compat.collections_abc.KeysView attribute), 8
- `__doc__` (pureyaml._compat.collections_abc.Mapping attribute), 8
- `__doc__` (pureyaml._compat.collections_abc.MappingView attribute), 9
- `__doc__` (pureyaml._compat.collections_abc.MutableMapping attribute), 9
- `__doc__` (pureyaml._compat.collections_abc.MutableSequence attribute), 10
- `__doc__` (pureyaml._compat.collections_abc.MutableSet attribute), 11
- `__doc__` (pureyaml._compat.collections_abc.Sequence attribute), 11
- `__doc__` (pureyaml._compat.collections_abc.Set attribute), 12
- `__doc__` (pureyaml._compat.collections_abc.Sized attribute), 13
- `__doc__` (pureyaml._compat.collections_abc.ValuesView attribute), 13
- `__doc__` (pureyaml._compat.singledispatch_helpers.ChainMap attribute), 14
- `__doc__` (pureyaml._compat.singledispatch_helpers.MappingProxyType attribute), 15
- `__doc__` (pureyaml._compat.singledispatch_helpers.Support attribute), 15
- `__doc__` (pureyaml.decoder.YAMLEncoder attribute), 29
- `__doc__` (pureyaml.encoder.SYMBOL attribute), 30
- `__doc__` (pureyaml.encoder.YAMLEncoder attribute), 30
- `__doc__` (pureyaml.exceptions.YAMLCastTypeError attribute), 31
- `__doc__` (pureyaml.exceptions.YAMLError attribute), 31
- `__doc__` (pureyaml.exceptions.YAMLStrictTypeError attribute), 31
- `__doc__` (pureyaml.exceptions.YAMLSyntaxError attribute), 31
- `__doc__` (pureyaml.exceptions.YAMLUnknownSyntaxError attribute), 31
- `__doc__` (pureyaml.grammar productions.YAMLProductions attribute), 16
- `__doc__` (pureyaml.grammar.tokens.TokenList attribute), 19
- `__doc__` (pureyaml.grammar.utils.fold attribute), 19
- `__doc__` (pureyaml.nodes.Binary attribute), 35
- `__doc__` (pureyaml.nodes.Bool attribute), 35
- `__doc__` (pureyaml.nodes.Collection attribute), 32
- `__doc__` (pureyaml.nodes.Doc attribute), 33
- `__doc__` (pureyaml.nodes.Docs attribute), 33
- `__doc__` (pureyaml.nodes.Float attribute), 34
- `__doc__` (pureyaml.nodes.Int attribute), 34
- `__doc__` (pureyaml.nodes.Map attribute), 33
- `__doc__` (pureyaml.nodes.Node attribute), 32
- `__doc__` (pureyaml.nodes.NodeVisitor attribute), 35
- `__doc__` (pureyaml.nodes.Null attribute), 34
- `__doc__` (pureyaml.nodes.Scalar attribute), 34
- `__doc__` (pureyaml.nodes.ScalarDispatch attribute), 35
- `__doc__` (pureyaml.nodes.Sequence attribute), 33
- `__doc__` (pureyaml.nodes.Str attribute), 34
- `__doc__` (pureyaml.parser.YAMLLexer attribute), 36
- `__doc__` (pureyaml.parser.YAMLParser attribute), 36
- `__doc__` (pureyaml.ply.cpp.Macro attribute), 20
- `__doc__` (pureyaml.ply.cpp.Preprocessor attribute), 20
- `__doc__` (pureyaml.ply.lex.LexError attribute), 21
- `__doc__` (pureyaml.ply.lex.LexToken attribute), 21
- `__doc__` (pureyaml.ply.lex.Lexer attribute), 22

`__doc__` (pureyaml.ply.lex.LexerReflect attribute), 22
`__doc__` (pureyaml.ply.lex.NullLogger attribute), 23
`__doc__` (pureyaml.ply.lex.PlyLogger attribute), 23
`__doc__` (pureyaml.ply.yacc.Grammar attribute), 23
`__doc__` (pureyaml.ply.yacc.GrammarError attribute), 24
`__doc__` (pureyaml.ply.yacc.LALRError attribute), 24
`__doc__` (pureyaml.ply.yacc.LRGeneratedTable attribute), 24
`__doc__` (pureyaml.ply.yacc.LRItem attribute), 25
`__doc__` (pureyaml.ply.yacc.LRParser attribute), 25
`__doc__` (pureyaml.ply.yacc.LRTable attribute), 25
`__doc__` (pureyaml.ply.yacc.MiniProduction attribute), 26
`__doc__` (pureyaml.ply.yacc.NullLogger attribute), 26
`__doc__` (pureyaml.ply.yacc.ParserReflect attribute), 26
`__doc__` (pureyaml.ply.yacc.PlyLogger attribute), 27
`__doc__` (pureyaml.ply.yacc.Production attribute), 27
`__doc__` (pureyaml.ply.yacc.VersionError attribute), 28
`__doc__` (pureyaml.ply.yacc.YaccError attribute), 28
`__doc__` (pureyaml.ply.yacc.YaccProduction attribute), 28
`__doc__` (pureyaml.ply.yacc.YaccSymbol attribute), 28
`__eq__()` (pureyaml._compat.collections_abc.Mapping method), 8
`__eq__()` (pureyaml._compat.collections_abc.Set method), 12
`__eq__()` (pureyaml.nodes.Float method), 34
`__eq__()` (pureyaml.nodes.Map method), 33
`__eq__()` (pureyaml.nodes.Node method), 32
`__eq__()` (pureyaml.nodes.Scalar method), 34
`__ge__()` (pureyaml._compat.collections_abc.Set method), 12
`__ge__()` (pureyaml.nodes.Node method), 32
`__getattr__()` (pureyaml.ply.lex.NullLogger method), 23
`__getattr__()` (pureyaml.ply.yacc.NullLogger method), 26
`__getitem__()` (pureyaml._compat.collections_abc.Mapping method), 8
`__getitem__()` (pureyaml._compat.collections_abc.Sequence method), 11
`__getitem__()` (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
`__getitem__()` (pureyaml.ply.yacc.Grammar method), 24
`__getitem__()` (pureyaml.ply.yacc.Production method), 27
`__getitem__()` (pureyaml.ply.yacc.YaccProduction method), 28
`__getslice__()` (pureyaml.ply.yacc.YaccProduction method), 28
`__gt__()` (pureyaml._compat.collections_abc.Set method), 12
`__gt__()` (pureyaml.nodes.Node method), 32
`__gt__()` (pureyaml.nodes.Scalar method), 34
`__hash__` (pureyaml._compat.collections_abc.Mapping attribute), 8
`__hash__` (pureyaml._compat.collections_abc.Set attribute), 12
`__hash__()` (pureyaml._compat.collections_abc.Hashable method), 6
`__iadd__()` (pureyaml._compat.collections_abc.MutableSequence method), 10
`__iand__()` (pureyaml._compat.collections_abc.MutableSet method), 11
`__init__()` (pureyaml._compat.collections_abc.MappingView method), 9
`__init__()` (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
`__init__()` (pureyaml._compat.singledispatch_helpers.MappingProxyType method), 15
`__init__()` (pureyaml.encoder.SYMBOL method), 30
`__init__()` (pureyaml.encoder.YAMLEncoder method), 30
`__init__()` (pureyaml.exceptions.YAMLCastTypeError method), 31
`__init__()` (pureyaml.exceptions.YAMLErrorException method), 31
`__init__()` (pureyaml.exceptions.YAMLStrictTypeError method), 31
`__init__()` (pureyaml.exceptions.YAMLSyntaxError method), 31
`__init__()` (pureyaml.nodes.Collection method), 32
`__init__()` (pureyaml.nodes.Node method), 32
`__init__()` (pureyaml.nodes.NodeVisitor method), 35
`__init__()` (pureyaml.nodes.Scalar method), 34
`__init__()` (pureyaml.parser.YAMLParser method), 36
`__init__()` (pureyaml.ply.cpp.Macro method), 20
`__init__()` (pureyaml.ply.cpp.Preprocessor method), 20
`__init__()` (pureyaml.ply.lex.LexError method), 21
`__init__()` (pureyaml.ply.lex.Lexer method), 22
`__init__()` (pureyaml.ply.lex.LexerReflect method), 22
`__init__()` (pureyaml.ply.lex.PlyLogger method), 23
`__init__()` (pureyaml.ply.yacc.Grammar method), 24
`__init__()` (pureyaml.ply.yacc.LRGeneratedTable method), 24
`__init__()` (pureyaml.ply.yacc.LRItem method), 25
`__init__()` (pureyaml.ply.yacc.LRParser method), 25
`__init__()` (pureyaml.ply.yacc.LRTable method), 26
`__init__()` (pureyaml.ply.yacc.MiniProduction method), 26
`__init__()` (pureyaml.ply.yacc.ParserReflect method), 26
`__init__()` (pureyaml.ply.yacc.PlyLogger method), 27
`__init__()` (pureyaml.ply.yacc.Production method), 27
`__init__()` (pureyaml.ply.yacc.YaccProduction method), 28
`__ior__()` (pureyaml._compat.collections_abc.MutableSet method), 11
`__isub__()` (pureyaml._compat.collections_abc.MutableSet method), 11

method), 11

`__iter__()` (pureyaml._compat.collections_abc.ItemsView method), 7

`__iter__()` (pureyaml._compat.collections_abc.Iterable method), 7

`__iter__()` (pureyaml._compat.collections_abc.Iterator method), 8

`__iter__()` (pureyaml._compat.collections_abc.KeysView method), 8

`__iter__()` (pureyaml._compat.collections_abc.Sequence method), 11

`__iter__()` (pureyaml._compat.collections_abc.ValuesView method), 13

`__iter__()` (pureyaml._compat singledispatch_helpers.ChainMap method), 14

`__iter__()` (pureyaml.ply.lex.Lexer method), 22

`__ixor__()` (pureyaml._compat.collections_abc.MutableSet method), 11

`__le__()` (pureyaml._compat.collections_abc.Set method), 12

`__le__()` (pureyaml.nodes.Node method), 32

`__len__()` (pureyaml._compat.collections_abc.MappingView method), 9

`__len__()` (pureyaml._compat.collections_abc.Sized method), 13

`__len__()` (pureyaml._compat singledispatch_helpers.ChainMap method), 14

`__len__()` (pureyaml.nodes.Scalar method), 34

`__len__()` (pureyaml.ply.yacc.Grammar method), 24

`__len__()` (pureyaml.ply.yacc.Production method), 27

`__len__()` (pureyaml.ply.yacc.YaccProduction method), 28

`__lt__()` (pureyaml._compat.collections_abc.Set method), 12

`__lt__()` (pureyaml.nodes.Node method), 32

`__metaclass__` (pureyaml._compat.collections_abc.Callable attribute), 6

`__metaclass__` (pureyaml._compat.collections_abc.Container attribute), 6

`__metaclass__` (pureyaml._compat.collections_abc.Hashable attribute), 6

`__metaclass__` (pureyaml._compat.collections_abc.Iterable attribute), 7

`__metaclass__` (pureyaml._compat.collections_abc.Sized attribute), 13

`__missing__()` (pureyaml._compat singledispatch_helpers.ChainMap method), 14

`__module__` (pureyaml._compat.NullHandler attribute), 5

`__module__` (pureyaml._compat.collections_abc attribute), 13

`__module__` (pureyaml._compat.collections_abc.Callable attribute), 6

`__module__` (pureyaml._compat.collections_abc.Container attribute), 6

`__module__` (pureyaml._compat.collections_abc.Hashable attribute), 6

`__module__` (pureyaml._compat.collections_abc.ItemsView attribute), 7

`__module__` (pureyaml._compat.collections_abc.Iterable attribute), 7

`__module__` (pureyaml._compat.collections_abc.Iterator attribute), 8

`__module__` (pureyaml._compat.collections_abc.KeysView attribute), 8

`__module__` (pureyaml._compat.collections_abc.Mapping attribute), 8

`__module__` (pureyaml._compat.collections_abc.MappingView attribute), 9

`__module__` (pureyaml._compat.collections_abc.MutableMapping attribute), 9

`__module__` (pureyaml._compat.collections_abc.MutableSequence attribute), 10

`__module__` (pureyaml._compat.collections_abc.MutableSet attribute), 11

`__module__` (pureyaml._compat.collections_abc.Sequence attribute), 11

`__module__` (pureyaml._compat.collections_abc.Set attribute), 12

`__module__` (pureyaml._compat.collections_abc.Sized attribute), 13

`__module__` (pureyaml._compat.collections_abc.ValuesView attribute), 13

`__module__` (pureyaml._compat singledispatch_helpers.ChainMap attribute), 14

`__module__` (pureyaml._compat singledispatch_helpers.MappingProxyType attribute), 15

`__module__` (pureyaml._compat singledispatch_helpers.Support attribute), 15

`__module__` (pureyaml.decoder.YAMLEncoder attribute), 29

`__module__` (pureyaml.encoder.SYMBOL attribute), 30

`__module__` (pureyaml.encoder.YAMLEncoder attribute), 30

`__module__` (pureyaml.exceptions.YAMLCastTypeError attribute), 31

`__module__` (pureyaml.exceptions.YAMLError attribute), 31

`__module__` (pureyaml.exceptions.YAMLStrictTypeError attribute), 31

`__module__` (pureyaml.exceptions.YAMLSyntaxError attribute), 31

`__module__` (pureyaml.exceptions.YAMLUnknownSyntaxError attribute), 31

`__module__` (pureyaml.grammar.productions.YAMLProductions attribute), 16

`__module__` (pureyaml.grammar.tokens.TokenList attribute), 19

- `__module__` (pureyaml.grammar.utils.fold attribute), 19
- `__module__` (pureyaml.nodes.Binary attribute), 35
- `__module__` (pureyaml.nodes.Bool attribute), 35
- `__module__` (pureyaml.nodes.Collection attribute), 32
- `__module__` (pureyaml.nodes.Doc attribute), 33
- `__module__` (pureyaml.nodes.Docs attribute), 33
- `__module__` (pureyaml.nodes.Float attribute), 34
- `__module__` (pureyaml.nodes.Int attribute), 34
- `__module__` (pureyaml.nodes.Map attribute), 33
- `__module__` (pureyaml.nodes.Node attribute), 32
- `__module__` (pureyaml.nodes.NodeVisitor attribute), 35
- `__module__` (pureyaml.nodes.Null attribute), 34
- `__module__` (pureyaml.nodes.Scalar attribute), 34
- `__module__` (pureyaml.nodes.ScalarDispatch attribute), 35
- `__module__` (pureyaml.nodes.Sequence attribute), 33
- `__module__` (pureyaml.nodes.Str attribute), 34
- `__module__` (pureyaml.parser.YAMLLexer attribute), 36
- `__module__` (pureyaml.parser.YAMLParser attribute), 36
- `__module__` (pureyaml.ply.cpp.Macro attribute), 20
- `__module__` (pureyaml.ply.cpp.Preprocessor attribute), 20
- `__module__` (pureyaml.ply.lex.LexError attribute), 21
- `__module__` (pureyaml.ply.lex.LexToken attribute), 21
- `__module__` (pureyaml.ply.lex.Lexer attribute), 22
- `__module__` (pureyaml.ply.lex.LexerReflect attribute), 22
- `__module__` (pureyaml.ply.lex.NullLogger attribute), 23
- `__module__` (pureyaml.ply.lex.PlyLogger attribute), 23
- `__module__` (pureyaml.ply.yacc.Grammar attribute), 24
- `__module__` (pureyaml.ply.yacc.GrammarError attribute), 24
- `__module__` (pureyaml.ply.yacc.LALRError attribute), 24
- `__module__` (pureyaml.ply.yacc.LRGeneratedTable attribute), 24
- `__module__` (pureyaml.ply.yacc.LRItem attribute), 25
- `__module__` (pureyaml.ply.yacc.LRParser attribute), 25
- `__module__` (pureyaml.ply.yacc.LRTable attribute), 26
- `__module__` (pureyaml.ply.yacc.MiniProduction attribute), 26
- `__module__` (pureyaml.ply.yacc.NullLogger attribute), 26
- `__module__` (pureyaml.ply.yacc.ParserReflect attribute), 26
- `__module__` (pureyaml.ply.yacc.PlyLogger attribute), 27
- `__module__` (pureyaml.ply.yacc.Production attribute), 27
- `__module__` (pureyaml.ply.yacc.VersionError attribute), 28
- `__module__` (pureyaml.ply.yacc.YaccError attribute), 28
- `__module__` (pureyaml.ply.yacc.YaccProduction attribute), 28
- `__module__` (pureyaml.ply.yacc.YaccSymbol attribute), 28
- `__ne__()` (pureyaml._compat.collections_abc.Mapping method), 8
- `__ne__()` (pureyaml._compat.collections_abc.Set method), 12
- `__ne__()` (pureyaml.nodes.Node method), 32
- `__new__()` (pureyaml.grammar.utils.fold static method), 19
- `__new__()` (pureyaml.nodes.ScalarDispatch static method), 35
- `__next__()` (pureyaml.ply.lex.Lexer method), 22
- `__nonzero__()` (pureyaml.ply.yacc.Production method), 27
- `__or__()` (pureyaml._compat.collections_abc.Set method), 12
- `__repr__()` (pureyaml._compat.collections_abc.MappingView method), 9
- `__repr__()` (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
- `__repr__()` (pureyaml.encoder.SYMBOL method), 30
- `__repr__()` (pureyaml.nodes.Node method), 32
- `__repr__()` (pureyaml.ply.lex.LexToken method), 21
- `__repr__()` (pureyaml.ply.yacc.LRItem method), 25
- `__repr__()` (pureyaml.ply.yacc.MiniProduction method), 26
- `__repr__()` (pureyaml.ply.yacc.Production method), 27
- `__repr__()` (pureyaml.ply.yacc.YaccSymbol method), 28
- `__reversed__()` (pureyaml._compat.collections_abc.Sequence method), 11
- `__setitem__()` (pureyaml._compat.collections_abc.MutableMapping method), 9
- `__setitem__()` (pureyaml._compat.collections_abc.MutableSequence method), 10
- `__setitem__()` (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
- `__setitem__()` (pureyaml.ply.yacc.YaccProduction method), 28
- `__str__()` (pureyaml.exceptions.YAMLError method), 31
- `__str__()` (pureyaml.exceptions.YAMLStrictTypeError method), 31
- `__str__()` (pureyaml.exceptions.YAMLSyntaxError method), 31
- `__str__()` (pureyaml.ply.lex.LexToken method), 21
- `__str__()` (pureyaml.ply.yacc.LRItem method), 25
- `__str__()` (pureyaml.ply.yacc.MiniProduction method), 26
- `__str__()` (pureyaml.ply.yacc.Production method), 27
- `__str__()` (pureyaml.ply.yacc.YaccSymbol method), 28
- `__sub__()` (pureyaml._compat.collections_abc.Set method), 12
- `__subclasshook__()` (pureyaml._compat.collections_abc.Callable class method), 6
- `__subclasshook__()` (pureyaml._compat.collections_abc.Container class method), 6

<code>__subclasshook__()</code> (pureyaml._compat.collections_abc.Hashable class method), 6	<code>__weakref__</code> (pureyaml.ply.yacc.PlyLogger attribute), 27
<code>__subclasshook__()</code> (pureyaml._compat.collections_abc.Iterable class method), 7	<code>__weakref__</code> (pureyaml.ply.yacc.Production attribute), 27
<code>__subclasshook__()</code> (pureyaml._compat.collections_abc.Iterator class method), 8	<code>__weakref__</code> (pureyaml.ply.yacc.YaccError attribute), 28
<code>__subclasshook__()</code> (pureyaml._compat.collections_abc.Sized class method), 13	<code>__xor__()</code> (pureyaml._compat.collections_abc.Set method), 12
<code>__weakref__</code> (pureyaml._compat.collections_abc.Callable attribute), 6	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Callable attribute), 6
<code>__weakref__</code> (pureyaml._compat.collections_abc.Container attribute), 6	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Container attribute), 6
<code>__weakref__</code> (pureyaml._compat.collections_abc.Hashable attribute), 6	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Hashable attribute), 7
<code>__weakref__</code> (pureyaml._compat.collections_abc.Hashable attribute), 7	<code>_abc_cache</code> (pureyaml._compat.collections_abc.ItemsView attribute), 7
<code>__weakref__</code> (pureyaml._compat.collections_abc.Iterable attribute), 7	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Iterable attribute), 7
<code>__weakref__</code> (pureyaml._compat.collections_abc.Sized attribute), 13	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Iterator attribute), 8
<code>__weakref__</code> (pureyaml._compat singledispatch_helpers.SupportsAbs attribute), 15	<code>_abc_cache</code> (pureyaml._compat.collections_abc.KeysView attribute), 8
<code>__weakref__</code> (pureyaml.exceptions.YAMLCastTypeError attribute), 31	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Mapping attribute), 8
<code>__weakref__</code> (pureyaml.exceptions.YAMLError attribute), 31	<code>_abc_cache</code> (pureyaml._compat.collections_abc.MappingView attribute), 9
<code>__weakref__</code> (pureyaml.exceptions.YAMLStrictTypeError attribute), 31	<code>_abc_cache</code> (pureyaml._compat.collections_abc.MutableMapping attribute), 9
<code>__weakref__</code> (pureyaml.exceptions.YAMLSyntaxError attribute), 31	<code>_abc_cache</code> (pureyaml._compat.collections_abc.MutableSequence attribute), 10
<code>__weakref__</code> (pureyaml.exceptions.YAMLUnknownSyntaxError attribute), 31	<code>_abc_cache</code> (pureyaml._compat.collections_abc.MutableSet attribute), 11
<code>__weakref__</code> (pureyaml.grammar.tokens.TokenList attribute), 19	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Sequence attribute), 11
<code>__weakref__</code> (pureyaml.grammar.utils.fold attribute), 19	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Set attribute), 12
<code>__weakref__</code> (pureyaml.nodes.Node attribute), 32	<code>_abc_cache</code> (pureyaml._compat.collections_abc.Sized attribute), 13
<code>__weakref__</code> (pureyaml.nodes.NodeVisitor attribute), 35	<code>_abc_cache</code> (pureyaml._compat.collections_abc.ValuesView attribute), 13
<code>__weakref__</code> (pureyaml.nodes.ScalarDispatch attribute), 35	<code>_abc_cache</code> (pureyaml._compat singledispatch_helpers.ChainMap attribute), 14
<code>__weakref__</code> (pureyaml.ply.cpp.Macro attribute), 20	<code>_abc_cache</code> (pureyaml.nodes.Collection attribute), 32
<code>__weakref__</code> (pureyaml.ply.cpp.Preprocessor attribute), 20	<code>_abc_cache</code> (pureyaml.nodes.Doc attribute), 33
<code>__weakref__</code> (pureyaml.ply.lex.LexError attribute), 21	<code>_abc_cache</code> (pureyaml.nodes Docs attribute), 33
<code>__weakref__</code> (pureyaml.ply.lex.LexToken attribute), 21	<code>_abc_cache</code> (pureyaml.nodes.Map attribute), 33
<code>__weakref__</code> (pureyaml.ply.lex.LexerReflect attribute), 22	<code>_abc_cache</code> (pureyaml.nodes.Sequence attribute), 33
<code>__weakref__</code> (pureyaml.ply.lex.NullLogger attribute), 23	<code>_abc_negative_cache</code> (pureyaml._compat.collections_abc.Callable attribute), 6
<code>__weakref__</code> (pureyaml.ply.lex.PlyLogger attribute), 23	<code>_abc_negative_cache</code> (pureyaml._compat.collections_abc.Container attribute), 6
<code>__weakref__</code> (pureyaml.ply.yacc.Grammar attribute), 24	<code>_abc_negative_cache</code> (pureyaml._compat.collections_abc.Hashable attribute), 7
<code>__weakref__</code> (pureyaml.ply.yacc.LRItem attribute), 25	<code>_abc_negative_cache</code> (pureyaml._compat.collections_abc.ItemsView attribute), 7
<code>__weakref__</code> (pureyaml.ply.yacc.LRTable attribute), 26	<code>_abc_negative_cache</code> (pureyaml._compat.collections_abc.Iterable
<code>__weakref__</code> (pureyaml.ply.yacc.Miniproduction attribute), 26	
<code>__weakref__</code> (pureyaml.ply.yacc.NullLogger attribute), 26	
<code>__weakref__</code> (pureyaml.ply.yacc.ParserReflect attribute), 26	

attribute), 7
 _abc_negative_cache (pureyaml._compat.collections_abc.Iterator
 attribute), 8
 _abc_negative_cache (pureyaml._compat.collections_abc.KeysView
 attribute), 8
 _abc_negative_cache (pureyaml._compat.collections_abc.Mapping
 attribute), 8
 _abc_negative_cache (pureyaml._compat.collections_abc.MappingView
 attribute), 9
 _abc_negative_cache (pureyaml._compat.collections_abc.MutableMapping
 attribute), 9
 _abc_negative_cache (pureyaml._compat.collections_abc.MutableSequence
 attribute), 10
 _abc_negative_cache (pureyaml._compat.collections_abc.MutableSet
 attribute), 11
 _abc_negative_cache (pureyaml._compat.collections_abc.Sequence
 attribute), 11
 _abc_negative_cache (pureyaml._compat.collections_abc.Set
 attribute), 12
 _abc_negative_cache (pureyaml._compat.collections_abc.Sized
 attribute), 13
 _abc_negative_cache (pureyaml._compat.collections_abc.ValuesView
 attribute), 13
 _abc_negative_cache (pureyaml._compat singledispatch_helpers.ChainMap
 attribute), 14
 _abc_negative_cache (pureyaml.nodes.Collection attribute), 32
 _abc_negative_cache (pureyaml.nodes.Doc attribute), 33
 _abc_negative_cache (pureyaml.nodes.Docs attribute), 33
 _abc_negative_cache (pureyaml.nodes.Map attribute), 33
 _abc_negative_cache (pureyaml.nodes.Sequence attribute), 33
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Callable
 attribute), 6
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Container
 attribute), 6
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Hashable
 attribute), 7
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.ItemsView
 attribute), 7
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Iterable
 attribute), 7
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Iterator
 attribute), 8
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.KeysView
 attribute), 8
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Mapping
 attribute), 8
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.MappingView
 attribute), 9
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.MutableMapping
 attribute), 9
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.MutableSequence
 attribute), 10
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.MutableSet
 attribute), 11
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Sequence
 attribute), 11
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Set
 attribute), 12
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.Sized
 attribute), 13
 _abc_negative_cache_version
 (pureyaml._compat.collections_abc.ValuesView
 attribute), 13
 _abc_negative_cache_version
 (pureyaml._compat singledispatch_helpers.ChainMap
 attribute), 14
 _abc_negative_cache_version
 (pureyaml.nodes.Collection attribute), 32
 _abc_negative_cache_version
 (pureyaml.nodes.Doc attribute), 33
 _abc_negative_cache_version
 (pureyaml.nodes.Docs attribute), 33
 _abc_negative_cache_version
 (pureyaml.nodes.Map attribute), 33
 _abc_negative_cache_version
 (pureyaml.nodes.Sequence attribute), 33
 _abc_registry (pureyaml._compat.collections_abc.Callable
 attribute), 6
 _abc_registry (pureyaml._compat.collections_abc.Container
 attribute), 6
 _abc_registry (pureyaml._compat.collections_abc.Hashable
 attribute), 7
 _abc_registry (pureyaml._compat.collections_abc.ItemsView
 attribute), 7
 _abc_registry (pureyaml._compat.collections_abc.Iterable
 attribute), 7
 _abc_registry (pureyaml._compat.collections_abc.Iterator
 attribute), 8
 _abc_registry (pureyaml._compat.collections_abc.KeysView
 attribute), 8
 _abc_registry (pureyaml._compat.collections_abc.Mapping
 attribute), 8

- attribute), 9
 - _abc_registry (pureyaml._compat.collections_abc.MappingView attribute), 9
 - _abc_registry (pureyaml._compat.collections_abc.MutableMapping attribute), 9
 - _abc_registry (pureyaml._compat.collections_abc.MutableSequence attribute), 10
 - _abc_registry (pureyaml._compat.collections_abc.MutableSet attribute), 11
 - _abc_registry (pureyaml._compat.collections_abc.Sequence attribute), 11
 - _abc_registry (pureyaml._compat.collections_abc.Set attribute), 12
 - _abc_registry (pureyaml._compat.collections_abc.Sized attribute), 13
 - _abc_registry (pureyaml._compat.collections_abc.ValuesView attribute), 13
 - _abc_registry (pureyaml._compat singledispatch_helpers.ChainMap attribute), 14
 - _abc_registry (pureyaml.nodes.Collection attribute), 32
 - _abc_registry (pureyaml.nodes.Doc attribute), 33
 - _abc_registry (pureyaml.nodes Docs attribute), 33
 - _abc_registry (pureyaml.nodes.Map attribute), 33
 - _abc_registry (pureyaml.nodes.Sequence attribute), 33
 - _encode() (pureyaml.encoder.YAMLEncoder method), 30
 - _first() (pureyaml.ply.yacc.Grammar method), 24
 - _fold_replace() (pureyaml.grammar.utils.fold class method), 19
 - _form_master_re() (in module pureyaml.ply.lex), 23
 - _from_iterable() (pureyaml._compat.collections_abc.ItemsView class method), 7
 - _from_iterable() (pureyaml._compat.collections_abc.KeysView class method), 8
 - _from_iterable() (pureyaml._compat.collections_abc.Set class method), 12
 - _funcs_to_names() (in module pureyaml.ply.lex), 23
 - _ge_from_gt() (in module pureyaml._compat.total_ordering), 15
 - _ge_from_le() (in module pureyaml._compat.total_ordering), 15
 - _ge_from_lt() (in module pureyaml._compat.total_ordering), 15
 - _get_regex() (in module pureyaml.ply.lex), 23
 - _gt_from_ge() (in module pureyaml._compat.total_ordering), 15
 - _gt_from_le() (in module pureyaml._compat.total_ordering), 16
 - _gt_from_lt() (in module pureyaml._compat.total_ordering), 16
 - _hash() (pureyaml._compat.collections_abc.Set method), 12
 - _le_from_ge() (in module pureyaml._compat.total_ordering), 16
 - _le_from_gt() (in module pureyaml._compat.total_ordering), 16
 - _le_from_lt() (in module pureyaml._compat.total_ordering), 16
 - _lt_from_ge() (in module pureyaml._compat.total_ordering), 16
 - _lt_from_gt() (in module pureyaml._compat.total_ordering), 16
 - _lt_from_le() (in module pureyaml._compat.total_ordering), 16
 - _names_to_funcs() (in module pureyaml.ply.lex), 23
 - _re_fold_replace (pureyaml.grammar.utils.fold attribute), 19
 - _statetoken() (in module pureyaml.ply.lex), 23
 - _visit() (pureyaml.nodes.NodeVisitor method), 35
- ## A
- add() (pureyaml._compat.collections_abc.MutableSet method), 11
 - add_lalr_lookaheads() (pureyaml.ply.yacc.LRGeneratedTable method), 24
 - add_lookaheads() (pureyaml.ply.yacc.LRGeneratedTable method), 24
 - add_path() (pureyaml.ply.cpp.Preprocessor method), 20
 - add_production() (pureyaml.ply.yacc.Grammar method), 24
 - append() (pureyaml._compat.collections_abc.MutableSequence method), 10
- ## B
- begin() (pureyaml.ply.lex.Lexer method), 22
 - Binary (class in pureyaml.nodes), 35
 - bind() (pureyaml.ply.yacc.Miniproduction method), 26
 - bind() (pureyaml.ply.yacc.Production method), 28
 - bind_callables() (pureyaml.ply.yacc.LRTable method), 26
 - Bool (class in pureyaml.nodes), 34
 - build() (pureyaml.parser.YAMLLexer class method), 36
 - build_litems() (pureyaml.ply.yacc.Grammar method), 24
- ## C
- call_errorfunc() (in module pureyaml.ply.yacc), 28
 - ChainMap (class in pureyaml._compat.singledispatch_helpers), 14
 - clear() (pureyaml._compat.collections_abc.MutableMapping method), 9
 - clear() (pureyaml._compat.collections_abc.MutableSet method), 11
 - clear() (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
 - clone() (pureyaml.ply.lex.Lexer method), 22
 - collect_args() (pureyaml.ply.cpp.Preprocessor method), 20
 - Collection (class in pureyaml.nodes), 32
 - collections_abc (class in pureyaml._compat), 5

- collections_abc.Callable (class in pureyaml._compat), 5
 - collections_abc.Container (class in pureyaml._compat), 6
 - collections_abc.Hashable (class in pureyaml._compat), 6
 - collections_abc.ItemsView (class in pureyaml._compat), 7
 - collections_abc.Iterable (class in pureyaml._compat), 7
 - collections_abc.Iterator (class in pureyaml._compat), 7
 - collections_abc.KeysView (class in pureyaml._compat), 8
 - collections_abc.Mapping (class in pureyaml._compat), 8
 - collections_abc.MappingView (class in pureyaml._compat), 9
 - collections_abc.MutableMapping (class in pureyaml._compat), 9
 - collections_abc.MutableSequence (class in pureyaml._compat), 10
 - collections_abc.MutableSet (class in pureyaml._compat), 10
 - collections_abc.Sequence (class in pureyaml._compat), 11
 - collections_abc.Set (class in pureyaml._compat), 12
 - collections_abc.Sized (class in pureyaml._compat), 13
 - collections_abc.ValuesView (class in pureyaml._compat), 13
 - compute_first() (pureyaml.ply.yacc.Grammar method), 24
 - compute_follow() (pureyaml.ply.yacc.Grammar method), 24
 - compute_follow_sets() (pureyaml.ply.yacc.LRGeneratedTable method), 24
 - compute_lookback_includes() (pureyaml.ply.yacc.LRGeneratedTable method), 24
 - compute_nullable_nonterminals() (pureyaml.ply.yacc.LRGeneratedTable method), 24
 - compute_read_sets() (pureyaml.ply.yacc.LRGeneratedTable method), 25
 - copy() (pureyaml._compat.singledispatch_helpers.ChainMap method), 14
 - count() (pureyaml._compat.collections_abc.Sequence method), 11
 - CPP_INTEGER() (in module pureyaml.ply.cpp), 20
 - cpython_only() (pureyaml._compat.singledispatch_helpers.Support method), 15
 - createLock() (pureyaml._compat.NullHandler method), 5
 - critical() (pureyaml.ply.lex.PlyLogger method), 23
 - critical() (pureyaml.ply.yacc.PlyLogger method), 27
 - current_state() (pureyaml.ply.lex.Lexer method), 22
 - digraph() (in module pureyaml.ply.yacc), 28
 - disable_defaulted_states() (pureyaml.ply.yacc.LRParser method), 25
 - discard() (pureyaml._compat.collections_abc.MutableSet method), 11
 - Doc (class in pureyaml.nodes), 33
 - Docs (class in pureyaml.nodes), 32
 - dr_relation() (pureyaml.ply.yacc.LRGeneratedTable method), 25
 - dummy() (pureyaml._compat.singledispatch_helpers.Support method), 15
 - dump() (in module pureyaml), 5
 - dumps() (in module pureyaml), 5
- ## E
- emit() (pureyaml._compat.NullHandler method), 5
 - encode() (pureyaml.encoder.YAMLEncoder method), 30
 - errok() (in module pureyaml.ply.yacc), 28
 - errok() (pureyaml.ply.yacc.LRParser method), 25
 - error() (pureyaml.ply.cpp.Preprocessor method), 20
 - error() (pureyaml.ply.lex.PlyLogger method), 23
 - error() (pureyaml.ply.yacc.PlyLogger method), 27
 - error() (pureyaml.ply.yacc.YaccProduction method), 28
 - evalexpr() (pureyaml.ply.cpp.Preprocessor method), 20
 - expand_macros() (pureyaml.ply.cpp.Preprocessor method), 20
 - extend() (pureyaml._compat.collections_abc.MutableSequence method), 10
- ## F
- FALSE_VALUES (pureyaml.nodes.Bool attribute), 35
 - filter_section() (in module pureyaml.ply.ygen), 29
 - find_column() (in module pureyaml.grammar.utils), 19
 - find_nonterminal_transitions() (pureyaml.ply.yacc.LRGeneratedTable method), 25
 - find_unreachable() (pureyaml.ply.yacc.Grammar method), 24
 - Float (class in pureyaml.nodes), 34
 - fold (class in pureyaml.grammar.utils), 19
 - format_result() (in module pureyaml.ply.yacc), 29
 - format_stack_entry() (in module pureyaml.ply.yacc), 29
 - from_decoded() (pureyaml.nodes.Binary class method), 35
 - fromkeys() (pureyaml._compat.singledispatch_helpers.ChainMap class method), 15
- ## G
- generic_visit() (pureyaml.nodes.NodeVisitor method), 35
 - get() (pureyaml._compat.collections_abc.Mapping method), 9
 - get() (pureyaml._compat.singledispatch_helpers.ChainMap method), 15
 - get_all() (pureyaml.ply.lex.LexerReflect method), 22

`get_all()` (pureyaml.ply.yacc.ParserReflect method), 26
`get_cache_token()` (in module pureyaml._compat.singledispatch_helpers), 15
`get_caller_module_dict()` (in module pureyaml.ply.lex), 23
`get_caller_module_dict()` (in module pureyaml.ply.yacc), 29
`get_error_func()` (pureyaml.ply.yacc.ParserReflect method), 26
`get_literals()` (pureyaml.ply.lex.LexerReflect method), 22
`get_pfunctions()` (pureyaml.ply.yacc.ParserReflect method), 26
`get_precedence()` (pureyaml.ply.yacc.ParserReflect method), 27
`get_rules()` (pureyaml.ply.lex.LexerReflect method), 22
`get_source_range()` (in module pureyaml.ply.ygen), 29
`get_start()` (pureyaml.ply.yacc.ParserReflect method), 27
`get_states()` (pureyaml.ply.lex.LexerReflect method), 22
`get_tokens()` (pureyaml.ply.lex.LexerReflect method), 22
`get_tokens()` (pureyaml.ply.yacc.ParserReflect method), 27
Grammar (class in pureyaml.ply.yacc), 23
GrammarError, 24
`group_lines()` (pureyaml.ply.cpp.Preprocessor method), 20

H

`handle()` (pureyaml._compat.NullHandler method), 5

I

`include()` (pureyaml.ply.cpp.Preprocessor method), 20
`index()` (pureyaml._compat.collections_abc.Sequence method), 12
`infinite_cycles()` (pureyaml.ply.yacc.Grammar method), 24
`info()` (pureyaml.ply.lex.PlyLogger method), 23
`info()` (pureyaml.ply.yacc.PlyLogger method), 27
`init_value()` (pureyaml.nodes.Binary method), 35
`init_value()` (pureyaml.nodes.Bool method), 35
`init_value()` (pureyaml.nodes.Collection method), 32
`init_value()` (pureyaml.nodes.Float method), 34
`init_value()` (pureyaml.nodes.Int method), 34
`init_value()` (pureyaml.nodes.Map method), 33
`init_value()` (pureyaml.nodes.Node method), 32
`init_value()` (pureyaml.nodes.Null method), 34
`init_value()` (pureyaml.nodes.Scalar method), 34
`init_value()` (pureyaml.nodes.Str method), 34
`input()` (pureyaml.ply.lex.Lexer method), 22
`insert()` (pureyaml._compat.collections_abc.MutableSequence method), 10
Int (class in pureyaml.nodes), 34
`is_float()` (in module pureyaml.encoder), 30

`isdisjoint()` (pureyaml._compat.collections_abc.Set method), 12
`items()` (pureyaml._compat.collections_abc.Mapping method), 9
`iter_map_items()` (pureyaml.encoder.YAMLEncoder method), 30
`iterencode()` (pureyaml.encoder.YAMLEncoder method), 30
`iteritems()` (pureyaml._compat.collections_abc.Mapping method), 9
`iterkeys()` (pureyaml._compat.collections_abc.Mapping method), 9
`itervalues()` (pureyaml._compat.collections_abc.Mapping method), 9

K

`keys()` (pureyaml._compat.collections_abc.Mapping method), 9

L

LALRError, 24
`lex()` (in module pureyaml.ply.lex), 23
Lexer (class in pureyaml.ply.lex), 22
LexerReflect (class in pureyaml.ply.lex), 22
LexError, 21
`lexpos()` (pureyaml.ply.yacc.YaccProduction method), 28
`lexprobe()` (pureyaml.ply.cpp.Preprocessor method), 20
`lexspan()` (pureyaml.ply.yacc.YaccProduction method), 28
LexToken (class in pureyaml.ply.lex), 21
`lineno()` (pureyaml.ply.yacc.YaccProduction method), 28
`linespan()` (pureyaml.ply.yacc.YaccProduction method), 28
`load()` (in module pureyaml), 5
`loads()` (in module pureyaml), 5
`lr0_closure()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`lr0_goto()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`lr0_items()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`lr_item()` (pureyaml.ply.yacc.Production method), 28
`lr_parse_table()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
LRGeneratedTable (class in pureyaml.ply.yacc), 24
LRItem (class in pureyaml.ply.yacc), 25
LRParser (class in pureyaml.ply.yacc), 25
LRTable (class in pureyaml.ply.yacc), 25

M

Macro (class in pureyaml.ply.cpp), 20
`macro_expand_args()` (pureyaml.ply.cpp.Preprocessor method), 20

macro_prescan() (pureyaml.ply.cpp.Preprocessor method), 20

main() (in module pureyaml.ply.ygen), 29

Map (class in pureyaml.nodes), 33

map (pureyaml.nodes.ScalarDispatch attribute), 35

MappingProxyType (class in pureyaml._compat.singledispatch_helpers), 15

MiniProduction (class in pureyaml.ply.yacc), 26

msg_lines() (pureyaml.exceptions.YAMLSyntaxError method), 31

N

new_child() (pureyaml._compat.singledispatch_helpers.ChainMap method), 15

next() (pureyaml._compat.collections_abc.Iterator method), 8

next() (pureyaml.ply.lex.Lexer method), 22

Node (class in pureyaml.nodes), 32

node_encoder() (in module pureyaml.encoder), 30

NodeVisitor (class in pureyaml.nodes), 35

Null (class in pureyaml.nodes), 34

NullHandler (class in pureyaml._compat), 5

NullLogger (class in pureyaml.ply.lex), 22

NullLogger (class in pureyaml.ply.yacc), 26

P

p_collection() (pureyaml.grammar productions.YAMLProductions method), 16

p_doc() (pureyaml.grammar productions.YAMLProductions method), 16

p_doc__indent() (pureyaml.grammar productions.YAMLProductions method), 16

p_doc_scalar_collection_ignore() (pureyaml.grammar productions.YAMLProductions method), 17

p_docs__init() (pureyaml.grammar productions.YAMLProductions method), 17

p_docs__last() (pureyaml.grammar productions.YAMLProductions method), 17

p_error() (pureyaml.parser.YAMLParser method), 36

p_flow_collection() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_map__init() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_map__last() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_map_item() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_map_item_key() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_map_item_value() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_sequence__init() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_sequence__last() (pureyaml.grammar productions.YAMLProductions method), 17

p_flow_sequence_item() (pureyaml.grammar productions.YAMLProductions method), 17

p_ignore__indent_dedent() (pureyaml.grammar productions.YAMLProductions method), 17

p_map__init() (pureyaml.grammar productions.YAMLProductions method), 17

p_map__last() (pureyaml.grammar productions.YAMLProductions method), 17

p_map_item() (pureyaml.grammar productions.YAMLProductions method), 17

p_map_item__key_value__collection() (pureyaml.grammar productions.YAMLProductions method), 17

p_map_item__compact_scalar() (pureyaml.grammar productions.YAMLProductions method), 17

p_map_item__key__map_item_value__sequence_item__compact_collection() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_key() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_key__complex_key_scalar() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_value__flow_collection() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_value__scalar() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_value__scalar_indented() (pureyaml.grammar productions.YAMLProductions method), 18

p_map_item_value__sequence_no_indent() (pureyaml.grammar productions.YAMLProductions method), 18

p_scalar() (pureyaml.grammar productions.YAMLProductions method), 18

p_scalar__doublequote() (pureyaml.grammar productions.YAMLProductions method), 18

p_scalar__explicit_cast() (pureyaml.grammar productions.YAMLProductions method), 18

p_scalar__folded() (pureyaml.grammar productions.YAMLProductions method), 18

p_scalar__indented_flow() (pureyaml.grammar productions.YAMLProductions method), 18

`p_scalar__literal()` (pureyaml.grammar.productions.YAMLProductions method), 15
 method), 18
`p_scalar__quote_empty()`
 (pureyaml.grammar.productions.YAMLProductions method), 18
`p_scalar__singlequote()` (pureyaml.grammar.productions.YAMLProductions method), 18
`p_scalar__string_indented_multi_line()`
 (pureyaml.grammar.productions.YAMLProductions method), 18
`p_scalar_group()` (pureyaml.grammar.productions.YAMLProductions method), 18
`p_sequence__init()` (pureyaml.grammar.productions.YAMLProductions method), 18
`p_sequence__last()` (pureyaml.grammar.productions.YAMLProductions method), 18
`p_sequence_item__collection()`
 (pureyaml.grammar.productions.YAMLProductions method), 19
`p_sequence_item__flow_collection()`
 (pureyaml.grammar.productions.YAMLProductions method), 19
`p_sequence_item__scalar()`
 (pureyaml.grammar.productions.YAMLProductions method), 19
`parents` (pureyaml._compat.singledispatch_helpers.ChainMap attribute), 15
`parse()` (pureyaml.parser.YAMLParser method), 36
`parse()` (pureyaml.ply.cpp.Preprocessor method), 20
`parse()` (pureyaml.ply.yacc.LRParser method), 25
`parse_grammar()` (in module pureyaml.ply.yacc), 29
`parsedebug()` (pureyaml.parser.YAMLParser method), 36
`parsedebug()` (pureyaml.ply.yacc.LRParser method), 25
`parsegen()` (pureyaml.ply.cpp.Preprocessor method), 20
`parseopt()` (pureyaml.ply.yacc.LRParser method), 25
`parseopt_notrack()` (pureyaml.ply.yacc.LRParser method), 25
`ParserReflect` (class in pureyaml.ply.yacc), 26
`pickle_table()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`PlyLogger` (class in pureyaml.ply.lex), 23
`PlyLogger` (class in pureyaml.ply.yacc), 27
`pop()` (pureyaml._compat.collections_abc.MutableMapping method), 9
`pop()` (pureyaml._compat.collections_abc.MutableSequence method), 10
`pop()` (pureyaml._compat.collections_abc.MutableSet method), 11
`pop()` (pureyaml._compat.singledispatch_helpers.ChainMap method), 15
`pop_state()` (pureyaml.ply.lex.Lexer method), 22
`popitem()` (pureyaml._compat.collections_abc.MutableMapping method), 10
`popitem()` (pureyaml._compat.singledispatch_helpers.ChainMap

Preprocessor (class in pureyaml.ply.cpp), 20
Production (class in pureyaml.ply.yacc), 27
pureyaml (module), 5
pureyaml._compat (module), 5
pureyaml._compat.singledispatch (module), 14
pureyaml._compat.singledispatch_helpers (module), 14
pureyaml._compat.total_ordering (module), 15
pureyaml.decoder (module), 29
pureyaml.encoder (module), 30
pureyaml.exceptions (module), 31
pureyaml.grammar (module), 16
pureyaml.grammar._parsetab (module), 16
pureyaml.grammar.productions (module), 16
pureyaml.grammar.tokens (module), 19
pureyaml.grammar.utils (module), 19
pureyaml.nodes (module), 32
pureyaml.parser (module), 36
pureyaml.ply (module), 20
pureyaml.ply.cpp (module), 20
pureyaml.ply.ctokens (module), 21
pureyaml.ply.lex (module), 21
pureyaml.ply.yacc (module), 23
pureyaml.ply.ygen (module), 29
push_state() (pureyaml.ply.lex.Lexer method), 22

R

`re_dispatch` (pureyaml.nodes.ScalarDispatch attribute), 35
`read_pickle()` (pureyaml.ply.yacc.LRTable method), 26
`read_table()` (pureyaml.ply.yacc.LRTable method), 26
`reads_relation()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`readtab()` (pureyaml.ply.lex.Lexer method), 22
`recursive_repr()` (in module pureyaml._compat.singledispatch_helpers), 15
`reduced` (pureyaml.ply.yacc.Production attribute), 28
`remove()` (pureyaml._compat.collections_abc.MutableSequence method), 10
`remove()` (pureyaml._compat.collections_abc.MutableSet method), 11
`repr_value()` (pureyaml.nodes.Node method), 32
`restart()` (in module pureyaml.ply.yacc), 29
`restart()` (pureyaml.ply.yacc.LRParser method), 25
`reverse()` (pureyaml._compat.collections_abc.MutableSequence method), 10
`rightmost_terminal()` (in module pureyaml.ply.yacc), 29
`rollback_lexpos()` (in module pureyaml.grammar.utils), 19
`runmain()` (in module pureyaml.ply.lex), 23

S

`Scalar` (class in pureyaml.nodes), 33

- ScalarDispatch (class in pureyaml.nodes), 35
- Sequence (class in pureyaml.nodes), 33
- set_defaulted_states() (pureyaml.ply.yacc.LRParser method), 25
- set_lineno() (pureyaml.ply.yacc.YaccProduction method), 28
- set_precedence() (pureyaml.ply.yacc.Grammar method), 24
- set_start() (pureyaml.ply.yacc.Grammar method), 24
- setdefault() (pureyaml._compat.collections_abc.MutableMapping method), 10
- signature() (pureyaml.ply.yacc.ParserReflect method), 27
- singledispatch() (in module pureyaml._compat), 13
- singledispatch() (in module pureyaml._compat.singledispatch), 14
- skip() (pureyaml.ply.lex.Lexer method), 22
- stack (pureyaml.encoder.YAMLEncoder attribute), 30
- Str (class in pureyaml.nodes), 34
- strict() (in module pureyaml.grammar.utils), 19
- Support (class in pureyaml._compat.singledispatch_helpers), 15
- SYMBOL (class in pureyaml.encoder), 30
- ## T
- t_ANY_error() (pureyaml.parser.YAMLLexer method), 36
- t_COMMENT() (in module pureyaml.ply.ctokens), 21
- t_CPP_CHAR() (in module pureyaml.ply.cpp), 21
- t_CPP_COMMENT1() (in module pureyaml.ply.cpp), 21
- t_CPP_COMMENT2() (in module pureyaml.ply.cpp), 21
- t_CPP_INTEGER() (in module pureyaml.ply.cpp), 21
- t_CPP_STRING() (in module pureyaml.ply.cpp), 21
- t_CPP_WS() (in module pureyaml.ply.cpp), 21
- t_CPPCOMMENT() (in module pureyaml.ply.ctokens), 21
- t_error() (in module pureyaml.ply.cpp), 21
- TOKEN() (in module pureyaml.ply.lex), 23
- Token() (in module pureyaml.ply.lex), 23
- token() (in module pureyaml.ply.yacc), 29
- token() (pureyaml.ply.cpp.Preprocessor method), 20
- token() (pureyaml.ply.lex.Lexer method), 22
- tokenize() (pureyaml.parser.YAMLLexer class method), 36
- tokenize() (pureyaml.parser.YAMLParser method), 36
- tokenize() (pureyaml.ply.cpp.Preprocessor method), 20
- TokenList (class in pureyaml.grammar.tokens), 19
- tokens (pureyaml.grammar.tokens.TokenList attribute), 19
- tokenstrip() (pureyaml.ply.cpp.Preprocessor method), 20
- total_ordering() (in module pureyaml._compat), 13
- total_ordering() (in module pureyaml._compat.total_ordering), 16
- traverse() (in module pureyaml.ply.yacc), 29
- trigraph() (in module pureyaml.ply.cpp), 21
- TRUE_VALUES (pureyaml.nodes.Bool attribute), 35
- type (pureyaml.nodes.Binary attribute), 35
- type (pureyaml.nodes.Bool attribute), 35
- type (pureyaml.nodes.Float attribute), 34
- type (pureyaml.nodes.Int attribute), 34
- type (pureyaml.nodes.Null attribute), 34
- type (pureyaml.nodes.Scalar attribute), 34
- type (pureyaml.nodes.Str attribute), 34
- ## U
- undef() (pureyaml.ply.cpp.Preprocessor method), 21
- undefined_symbols() (pureyaml.ply.yacc.Grammar method), 24
- unused_precedence() (pureyaml.ply.yacc.Grammar method), 24
- unused_rules() (pureyaml.ply.yacc.Grammar method), 24
- unused_terminals() (pureyaml.ply.yacc.Grammar method), 24
- update() (pureyaml._compat.collections_abc.MutableMapping method), 10
- ## V
- validate_all() (pureyaml.ply.lex.LexerReflect method), 22
- validate_all() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_error_func() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_literals() (pureyaml.ply.lex.LexerReflect method), 22
- validate_module() (pureyaml.ply.lex.LexerReflect method), 22
- validate_modules() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_pfunctions() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_precedence() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_rules() (pureyaml.ply.lex.LexerReflect method), 22
- validate_start() (pureyaml.ply.yacc.ParserReflect method), 27
- validate_tokens() (pureyaml.ply.lex.LexerReflect method), 22
- validate_tokens() (pureyaml.ply.yacc.ParserReflect method), 27
- values() (pureyaml._compat.collections_abc.Mapping method), 9
- VersionError, 28
- visit() (pureyaml.nodes.NodeVisitor method), 35
- visit_Binary() (pureyaml.decoder.YAMLEncoder method), 29
- visit_Binary() (pureyaml.encoder.YAMLEncoder method), 30

`visit_Bool()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Bool()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Doc()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Docs()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Float()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Float()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Int()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Int()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Map()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Map()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Null()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Null()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Scalar()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Scalar()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Sequence()` (pureyaml.decoder.YAMLEncoder method), 29
`visit_Sequence()` (pureyaml.encoder.YAMLEncoder method), 30
`visit_Str()` (pureyaml.decoder.YAMLEncoder method), 30
`visit_Str()` (pureyaml.encoder.YAMLEncoder method), 30

`YAMLParse` (class in pureyaml.parser), 36
`YAMLProductions` (class pureyaml.grammar.productions), 16
`YAMLStrictTypeError`, 31
`YAMLSyntaxError`, 31
`YAMLUnknownSyntaxError`, 31

W

`warning()` (pureyaml.ply.lex.PlyLogger method), 23
`warning()` (pureyaml.ply.yacc.PlyLogger method), 27
`write_table()` (pureyaml.ply.yacc.LRGeneratedTable method), 25
`writetab()` (pureyaml.ply.lex.Lexer method), 22

Y

`yacc()` (in module pureyaml.ply.yacc), 29
`YaccError`, 28
`YaccProduction` (class in pureyaml.ply.yacc), 28
`YaccSymbol` (class in pureyaml.ply.yacc), 28
`YAMLCastTypeError`, 31
`YAMLEncoder` (class in pureyaml.decoder), 29
`YAMLEncoder` (class in pureyaml.encoder), 30
`YAMLError`, 31
`YAMLLexer` (class in pureyaml.parser), 36