

:mod:`enum` --- Support for enumerations

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 1); [backlink](#)

Unknown interpreted text role "mod".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 4)

Unknown directive type "module".

```
.. module:: enum
   :synopsis: Implementation of an enumeration class.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 7)

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Ethan Furman <ethan@stoneleaf.us>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 8)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Barry Warsaw <barry@python.org>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 9)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Eli Bendersky <eliben@gmail.com>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 10)

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Ethan Furman <ethan@stoneleaf.us>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 12)

Unknown directive type "versionadded".

```
.. versionadded:: 3.4
```

Source code: `source:Lib/enum.py`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 14); [backlink](#)

Unknown interpreted text role "source".

Important

This page contains the API reference information. For tutorial information and discussion of more advanced topics, see

- [ref: Basic Tutorial <enum-basic-tutorial>](#)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 21); [backlink](#)

Unknown interpreted text role "ref".

- [ref: Advanced Tutorial <enum-advanced-tutorial>](#)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 22); [backlink](#)

Unknown interpreted text role "ref".

- [ref: Enum Cookbook <enum-cookbook>](#)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 23); [backlink](#)

Unknown interpreted text role "ref".

An enumeration:

- is a set of symbolic names (members) bound to unique values
- can be iterated over to return its members in definition order
- uses *call* syntax to return members by value
- uses *index* syntax to return members by name

Enumerations are created either by using `keyword: class` syntax, or by using function-call syntax:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 34); [backlink](#)

Unknown interpreted text role "keyword".

```
>>> from enum import Enum

>>> # class syntax
>>> class Color(Enum):
...     RED = 1
...     GREEN = 2
...     BLUE = 3

>>> # functional syntax
>>> Color = Enum('Color', ['RED', 'GREEN', 'BLUE'])
```

Even though we can use `keyword: class` syntax to create Enums, Enums are not normal Python classes. See [ref: How are Enums different? <enum-class-differences>](#) for more details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 48); [backlink](#)

Unknown interpreted text role "keyword".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 48); [backlink](#)

Unknown interpreted text role "ref".

Note

Nomenclature

- The class `class: Color` is an *enumeration* (or *enum*)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 54); [backlink](#)

Unknown interpreted text role "class".

- The attributes `attr: Color.RED`, `attr: Color.GREEN`, etc., are *enumeration members* (or *members*) and are functionally constants.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 55); [backlink](#)

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 55); [backlink](#)

Unknown interpreted text role "attr".

- The enum members have *names* and *values* (the name of `attr: Color.RED` is `RED`, the value of `attr: Color.BLUE` is `3`, etc.)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 57); [backlink](#)

Unknown interpreted text role "attr".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 57); [backlink](#)

Unknown interpreted text role "attr".

Module Contents

`:class:'EnumType'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 66); [backlink](#)

Unknown interpreted text role "class".

The `type` for `Enum` and its subclasses.

`:class:'Enum'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 70); [backlink](#)

Unknown interpreted text role "class".

Base class for creating enumerated constants.

`:class:'IntEnum'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 74); [backlink](#)

Unknown interpreted text role "class".

Base class for creating enumerated constants that are also subclasses of `:class:'int'`. (Notes)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 76); [backlink](#)

Unknown interpreted text role "class".

`:class:'StrEnum'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 79); [backlink](#)

Unknown interpreted text role "class".

Base class for creating enumerated constants that are also subclasses of `:class:'str'`. (Notes)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 81); [backlink](#)

Unknown interpreted text role "class".

`:class:'Flag'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 84); [backlink](#)

Unknown interpreted text role "class".

Base class for creating enumerated constants that can be combined using the bitwise operations without losing their `:class:'Flag'` membership.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 86); [backlink](#)

Unknown interpreted text role "class".

`:class:'IntFlag'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 89); [backlink](#)

Unknown interpreted text role "class".

Base class for creating enumerated constants that can be combined using the bitwise operators without losing their `:class:'IntFlag'` membership. `:class:'IntFlag'` members are also subclasses of `:class:'int'`. (Notes)

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 91); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 91); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 91); [backlink](#)

Unknown interpreted text role "class".

`:class:'EnumCheck'`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 95); [backlink](#)

Unknown interpreted text role "class".

An enumeration with the values `CONTINUOUS`, `NAMED_FLAGS`, and `UNIQUE`, for use with `:func:verify` to ensure various constraints are met by a given enumeration.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 97); [backlink](#)

Unknown interpreted text role "func".

`:class:FlagBoundary`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 101); [backlink](#)

Unknown interpreted text role "class".

An enumeration with the values `STRICT`, `CONFORM`, `EJECT`, and `KEEP` which allows for more fine-grained control over how invalid values are dealt with in an enumeration.

`:class:auto`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 107); [backlink](#)

Unknown interpreted text role "class".

Instances are replaced with an appropriate value for Enum members. `:class:StrEnum` defaults to the lower-cased version of the member name, while other Enums default to 1 and increase from there.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 109); [backlink](#)

Unknown interpreted text role "class".

`:func:~enum.property`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 113); [backlink](#)

Unknown interpreted text role "func".

Allows `:class:Enum` members to have attributes without conflicting with member names.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 115); [backlink](#)

Unknown interpreted text role "class".

`:func:unique`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 118); [backlink](#)

Unknown interpreted text role "func".

Enum class decorator that ensures only one name is bound to any one value.

`:func:verify`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 122); [backlink](#)

Unknown interpreted text role "func".

Enum class decorator that checks user-selectable constraints on an enumeration.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 128)

Unknown directive type "versionadded".

.. versionadded:: 3.6 ``Flag``, ``IntFlag``, ``auto``

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 129)

Unknown directive type "versionadded".

.. versionadded:: 3.11 ``StrEnum``, ``EnumCheck``, ``FlagBoundary``, ``property``

Data Types

`EnumType` is the `term metaclass` for `enum` enumerations. It is possible to subclass `EnumType` -- see [ref:Subclassing EnumType](#) <enumtype-examples> for details.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 139); [backlink](#)

Unknown interpreted text role "term".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 139); [backlink](#)

Unknown interpreted text role "ref".

EnumType is responsible for setting the correct `meth'__repr__', meth'__str__', meth'__format__',` and `meth'__reduce__` methods on the final *enum*, as well as creating the enum members, properly handling duplicates, providing iteration over the enum class, etc.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 143); [backlink](#)
Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 143); [backlink](#)
Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 143); [backlink](#)
Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 143); [backlink](#)
Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 148)
Unknown directive type "method".

```
.. method:: EnumType.__contains__(cls, member)

Returns ``True`` if member belongs to the ``cls``::

>>> some_var = Color.RED
>>> some_var in Color
True

.. note::

    In Python 3.12 it will be possible to check for member values and not
    just members; until then, a ``TypeError`` will be raised if a
    non-Enum-member is used in a containment check.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 162)
Unknown directive type "method".

```
.. method:: EnumType.__dir__(cls)

Returns ``['__class__', '__doc__', '__members__', '__module__']`` and the
names of the members in *cls*::

>>> dir(Color)
['BLUE', 'GREEN', 'RED', '__class__', '__contains__', '__doc__', '__getitem__', '__init_subclass__', '__iter__', '__len__', '...
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 170)
Unknown directive type "method".

```
.. method:: EnumType.__getattr__(cls, name)

Returns the Enum member in *cls* matching *name*, or raises an :exc:`AttributeError`::

>>> Color.GREEN
<Color.GREEN: 2>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 177)
Unknown directive type "method".

```
.. method:: EnumType.__getitem__(cls, name)

Returns the Enum member in *cls* matching *name*, or raises an :exc:`KeyError`::

>>> Color['BLUE']
<Color.BLUE: 3>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 184)
Unknown directive type "method".

```
.. method:: EnumType.__iter__(cls)

Returns each member in *cls* in definition order::

>>> list(Color)
[<Color.RED: 1>, <Color.GREEN: 2>, <Color.BLUE: 3>]
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 191)
Unknown directive type "method".

```
.. method:: EnumType.__len__(cls)

Returns the number of member in *cls*::

>>> len(Color)
3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 198)
Unknown directive type "method".

```
.. method:: EnumType.__reversed__(cls)

Returns each member in *cls* in reverse definition order::
```

```
>>> list(reversed(Color))
[<Color.BLUE: 3>, <Color.GREEN: 2>, <Color.RED: 1>]
```

Enum is the base class for all *enum* enumerations.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 210)

Unknown directive type "attribute".

```
.. attribute:: Enum.name

    The name used to define the ``Enum`` member::

    >>> Color.BLUE.name
    'BLUE'
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 217)

Unknown directive type "attribute".

```
.. attribute:: Enum.value

    The value given to the ``Enum`` member::

    >>> Color.RED.value
    1

.. note:: Enum member values

    Member values can be anything: :class:`int`, :class:`str`, etc.. If
    the exact value is unimportant you may use :class:`auto` instances and an
    appropriate value will be chosen for you. Care must be taken if you mix
    :class:`auto` with other values.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 231)

Unknown directive type "attribute".

```
.. attribute:: Enum.ignore_

    ``ignore`` is only used during creation and is removed from the
    enumeration once creation is complete.

    ``ignore`` is a list of names that will not become members, and whose
    names will also be removed from the completed enumeration. See
    :ref:`TimePeriod <enum-time-period>` for an example.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 240)

Unknown directive type "method".

```
.. method:: Enum.__call__(cls, value, names=None, *, module=None, qualname=None, type=None, start=1, boundary=None)

    This method is called in two different ways:

    * to look up an existing member:

        :cls: The enum class being called.
        :value: The value to lookup.

    * to use the ``cls`` enum to create a new enum:

        :cls: The enum class being called.
        :value: The name of the new Enum to create.
        :names: The names/values of the members for the new Enum.
        :module: The name of the module the new Enum is created in.
        :qualname: The actual location in the module where this Enum can be found.
        :type: A mix-in type for the new Enum.
        :start: The first integer value for the Enum (used by :class:`auto`)
        :boundary: How to handle out-of-range values from bit operations (:class:`Flag` only)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 260)

Unknown directive type "method".

```
.. method:: Enum.__dir__(self)

    Returns ``['__class__', '__doc__', '__module__', 'name', 'value']`` and
    any public methods defined on *self.__class__*:

    >>> from datetime import date
    >>> class Weekday(Enum):
    ...     MONDAY = 1
    ...     TUESDAY = 2
    ...     WEDNESDAY = 3
    ...     THURSDAY = 4
    ...     FRIDAY = 5
    ...     SATURDAY = 6
    ...     SUNDAY = 7
    ...     @classmethod
    ...     def today(cls):
    ...         print('today is %s' % cls(date.today().isoweekday()).name)
    >>> dir(Weekday.SATURDAY)
    ['__class__', '__doc__', '__eq__', '__hash__', '__module__', 'name', 'today', 'value']
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 280)

Unknown directive type "method".

```
.. method:: Enum.generate_next_value_(name, start, count, last_values)

    :name: The name of the member being defined (e.g. 'RED').
    :start: The start value for the Enum; the default is 1.
    :count: The number of members currently defined, not including this one.
    :last_values: A list of the previous values.

    A *staticmethod* that is used to determine the next value returned by
    :class:`auto`:

    >>> from enum import auto
    >>> class PowersOfThree(Enum):
```

```
...     @staticmethod
...     def _generate_next_value_(name, start, count, last_values):
...         return (count + 1) * 3
...     FIRST = auto()
...     SECOND = auto()
>>> PowersOfThree.SECOND.value
6
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 300)

Unknown directive type "method".

```
.. method:: Enum.__init_subclass__(cls, **kwargs)
```

A `*classmethod*` that is used to further configure subsequent subclasses. By default, does nothing.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 305)

Unknown directive type "method".

```
.. method:: Enum.__missing__(cls, value)
```

A `*classmethod*` for looking up values not found in `*cls*`. By default it does nothing, but can be overridden to implement custom search behavior::

```
>>> from enum import StrEnum
>>> class Build(StrEnum):
...     DEBUG = auto()
...     OPTIMIZED = auto()
...     @classmethod
...     def __missing__(cls, value):
...         value = value.lower()
...         for member in cls:
...             if member.value == value:
...                 return member
...         return None
>>> Build.DEBUG.value
'debug'
>>> Build('debug')
<Build.DEBUG: 'debug'>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 326)

Unknown directive type "method".

```
.. method:: Enum.__repr__(self)
```

Returns the string used for `*repr()*` calls. By default, returns the `*Enum*` name, member name, and value, but can be overridden::

```
>>> class OtherStyle(Enum):
...     ALTERNATE = auto()
...     OTHER = auto()
...     SOMETHING_ELSE = auto()
...     def __repr__(self):
...         cls_name = self.__class__.__name__
...         return f'{cls_name}.{self.name}'
>>> OtherStyle.ALTERNATE, str(OtherStyle.ALTERNATE), f'{OtherStyle.ALTERNATE}'
(OtherStyle.ALTERNATE, 'OtherStyle.ALTERNATE', 'OtherStyle.ALTERNATE')
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 341)

Unknown directive type "method".

```
.. method:: Enum.__str__(self)
```

Returns the string used for `*str()*` calls. By default, returns the `*Enum*` name and member name, but can be overridden::

```
>>> class OtherStyle(Enum):
...     ALTERNATE = auto()
...     OTHER = auto()
...     SOMETHING_ELSE = auto()
...     def __str__(self):
...         return f'{self.name}'
>>> OtherStyle.ALTERNATE, str(OtherStyle.ALTERNATE), f'{OtherStyle.ALTERNATE}'
(<OtherStyle.ALTERNATE: 1>, 'ALTERNATE', 'ALTERNATE')
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 355)

Unknown directive type "method".

```
.. method:: Enum.__format__(self)
```

Returns the string used for `*format()*` and `*f-string*` calls. By default, returns `:meth:`__str__`` returns, but can be overridden::

```
>>> class OtherStyle(Enum):
...     ALTERNATE = auto()
...     OTHER = auto()
...     SOMETHING_ELSE = auto()
...     def __format__(self, spec):
...         return f'{self.name}'
>>> OtherStyle.ALTERNATE, str(OtherStyle.ALTERNATE), f'{OtherStyle.ALTERNATE}'
(<OtherStyle.ALTERNATE: 1>, 'OtherStyle.ALTERNATE', 'ALTERNATE')
```

Note

Using `:class:`auto`` with `:class:`Enum`` results in integers of increasing value, starting with 1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 371); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 371); [backlink](#)

Unknown interpreted text role "class".

IntEnum is the same as *Enum*, but its members are also integers and can be used anywhere that an integer can be used. If any integer operation is performed with an *IntEnum* member, the resulting value loses its enumeration status.

```
>>> from enum import IntEnum
>>> class Numbers(IntEnum):
...     ONE = 1
...     TWO = 2
...     THREE = 3
>>> Numbers.THREE
<Numbers.THREE: 3>
>>> Numbers.ONE + Numbers.TWO
3
>>> Numbers.THREE + 5
8
>>> Numbers.THREE == 3
True
```

Note

Using `:class:'auto'` with `:class:'IntEnum'` results in integers of increasing value, starting with 1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 397); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 397); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 400)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.11 :meth:'__str__' is now :func:'int.__str__' to
better support the *replacement of existing constants* use-case.
:meth:'__format__' was already :func:'int.__format__' for that same reason.
```

StrEnum is the same as *Enum*, but its members are also strings and can be used in most of the same places that a string can be used. The result of any string operation performed on or with a *StrEnum* member is not part of the enumeration.

Note

There are places in the stdlib that check for an exact `:class:'str'` instead of a `:class:'str'` subclass (i.e. `type(unknown) == str` instead of `isinstance(str, unknown)`), and in those locations you will need to use `str(StrEnum.member)`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 411); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 411); [backlink](#)

Unknown interpreted text role "class".

Note

Using `:class:'auto'` with `:class:'StrEnum'` results in the lower-cased member name as the value.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 418); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 418); [backlink](#)

Unknown interpreted text role "class".

Note

`:meth:'__str__'` is `:func:'str.__str__'` to better support the *replacement of existing constants* use-case.
`:meth:'__format__'` is likewise `:func:'str.__format__'` for that same reason.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 421); [backlink](#)

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 421); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 421); [backlink](#)

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 421); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 425)

Unknown directive type "versionadded".

```
.. versionadded:: 3.11
```

Flag members support the bitwise operators $\&$ (*AND*), \mid (*OR*), \wedge (*XOR*), and \sim (*INVERT*); the results of those operators are members of the enumeration.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 433)

Unknown directive type "method".

```
.. method:: __contains__(self, value)

Returns *True* if value is in self::

    >>> from enum import Flag, auto
    >>> class Color(Flag):
    ...     RED = auto()
    ...     GREEN = auto()
    ...     BLUE = auto()
    >>> purple = Color.RED | Color.BLUE
    >>> white = Color.RED | Color.GREEN | Color.BLUE
    >>> Color.GREEN in purple
    False
    >>> Color.GREEN in white
    True
    >>> purple in white
    True
    >>> white in purple
    False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 453)

Unknown directive type "method".

```
.. method:: __iter__(self):

Returns all contained members::

    >>> list(Color.RED)
    [<Color.RED: 1>]
    >>> list(purple)
    [<Color.RED: 1>, <Color.BLUE: 4>]
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 462)

Unknown directive type "method".

```
.. method:: __len__(self):

Returns number of members in flag::

    >>> len(Color.GREEN)
    1
    >>> len(white)
    3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 471)

Unknown directive type "method".

```
.. method:: __bool__(self):

Returns *True* if any members in flag, *False* otherwise::

    >>> bool(Color.GREEN)
    True
    >>> bool(white)
    True
    >>> black = Color(0)
    >>> bool(black)
    False
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 483)

Unknown directive type "method".

```
.. method:: __or__(self, other)

Returns current flag binary or'ed with other::

    >>> Color.RED | Color.GREEN
    <Color.RED|GREEN: 3>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main [Doc] [library]enum.rst, line 490)

Unknown directive type "method".

```
.. method:: __and__(self, other)

Returns current flag binary and'ed with other::

    >>> purple & white
    <Color.RED|BLUE: 5>
    >>> purple & Color.GREEN
    <Color: 0>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 499)

Unknown directive type "method".

```
.. method:: __xor__(self, other)

    Returns current flag binary xor'ed with other::

    >>> purple ^ white
    <Color.GREEN: 2>
    >>> purple ^ Color.GREEN
    <Color.RED|GREEN|BLUE: 7>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 508)

Unknown directive type "method".

```
.. method:: __invert__(self):

    Returns all the flags in *type(self)* that are not in self::

    >>> ~white
    <Color: 0>
    >>> ~purple
    <Color.GREEN: 2>
    >>> ~Color.RED
    <Color.GREEN|BLUE: 6>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 519)

Unknown directive type "method".

```
.. method:: _numeric_repr_

    Function used to format any remaining unnamed numeric values. Default is
    the value's repr; common choices are :func:`hex` and :func:`oct`.
```

Note

Using :class:`auto` with :class:`Flag` results in integers that are powers of two, starting with 1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 526); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 526); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 529)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.11 The *repr()* of zero-valued flags has changed. It
    is now::

    >>> Color(0) # doctest: +SKIP
    <Color: 0>
```

IntFlag is the same as *Flag*, but its members are also integers and can be used anywhere that an integer can be used.

```
>>> from enum import IntFlag, auto
>>> class Color(IntFlag):
...     RED = auto()
...     GREEN = auto()
...     BLUE = auto()
>>> Color.RED & 2
<Color: 0>
>>> Color.RED | 2
<Color.RED|GREEN: 3>
```

If any integer operation is performed with an *IntFlag* member, the result is not an *IntFlag*:

```
>>> Color.RED + 2
3
```

If a *Flag* operation is performed with an *IntFlag* member and:

- the result is a valid *IntFlag*: an *IntFlag* is returned
- the result is not a valid *IntFlag*: the result depends on the *FlagBoundary* setting

The *repr()* of unnamed zero-valued flags has changed. It is now:

```
>>> Color(0)
<Color: 0>
```

Note

Using :class:`auto` with :class:`IntFlag` results in integers that are powers of two, starting with 1.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 568); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]enum.rst, line 568); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 571)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.11 :meth:`__str__` is now :func:`int.__str__` to
better support the *replacement of existing constants* use-case.
:meth:`__format__` was already :func:`int.__format__` for that same reason.
```

EnumCheck contains the options used by the `:func:`verify`` decorator to ensure various constraints; failed constraints result in a `:exc:`ValueError``.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 578); [backlink](#)

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 578); [backlink](#)

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 581)

Unknown directive type "attribute".

```
.. attribute:: UNIQUE

Ensure that each value has only one name::

>>> from enum import Enum, verify, UNIQUE
>>> @verify(UNIQUE)
... class Color(Enum):
...     RED = 1
...     GREEN = 2
...     BLUE = 3
...     CRIMSON = 1
Traceback (most recent call last):
...
ValueError: aliases found in <enum 'Color'>: CRIMSON -> RED
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 597)

Unknown directive type "attribute".

```
.. attribute:: CONTINUOUS

Ensure that there are no missing values between the lowest-valued member
and the highest-valued member::

>>> from enum import Enum, verify, CONTINUOUS
>>> @verify(CONTINUOUS)
... class Color(Enum):
...     RED = 1
...     GREEN = 2
...     BLUE = 5
Traceback (most recent call last):
...
ValueError: invalid enum 'Color': missing values 3, 4
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 612)

Unknown directive type "attribute".

```
.. attribute:: NAMED_FLAGS

Ensure that any flag groups/masks contain only named flags -- useful when
values are specified instead of being generated by :func:`auto`

>>> from enum import Flag, verify, NAMED_FLAGS
>>> @verify(NAMED_FLAGS)
... class Color(Flag):
...     RED = 1
...     GREEN = 2
...     BLUE = 4
...     WHITE = 15
...     NEON = 31
Traceback (most recent call last):
...
ValueError: invalid Flag 'Color': aliases WHITE and NEON are missing combined values of 0x18 [use enum.show_flag_values(value)]
```

Note

CONTINUOUS and NAMED_FLAGS are designed to work with integer-valued members.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 633)

Unknown directive type "versionadded".

```
.. versionadded:: 3.11
```

FlagBoundary controls how out-of-range values are handled in *Flag* and its subclasses.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 640)

Unknown directive type "attribute".

```
.. attribute:: STRICT

Out-of-range values cause a :exc:`ValueError` to be raised. This is the
default for :class:`Flag`:

>>> from enum import Flag, STRICT
>>> class StrictFlag(Flag, boundary=STRICT):
...     RED = auto()
...     GREEN = auto()
...     BLUE = auto()
>>> StrictFlag(2**2 + 2**4)
Traceback (most recent call last):
```

```
...
ValueError: <flag 'StrictFlag'> invalid value 20
given 0b0 10100
allowed 0b0 00111
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 657)

Unknown directive type "attribute".

```
.. attribute:: CONFORM

Out-of-range values have invalid values removed, leaving a valid *Flag*
value::

>>> from enum import Flag, CONFORM
>>> class ConformFlag(Flag, boundary=CONFORM):
...     RED = auto()
...     GREEN = auto()
...     BLUE = auto()
>>> ConformFlag(2**2 + 2**4)
<ConformFlag.BLUE: 4>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 670)

Unknown directive type "attribute".

```
.. attribute:: EJECT

Out-of-range values lose their *Flag* membership and revert to :class:`int`.
This is the default for :class:`IntFlag`::

>>> from enum import Flag, EJECT
>>> class EjectFlag(Flag, boundary=EJECT):
...     RED = auto()
...     GREEN = auto()
...     BLUE = auto()
>>> EjectFlag(2**2 + 2**4)
20
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 683)

Unknown directive type "attribute".

```
.. attribute:: KEEP

Out-of-range values are kept, and the *Flag* membership is kept. This is
used for some stdlib flags:

>>> from enum import Flag, KEEP
>>> class KeepFlag(Flag, boundary=KEEP):
...     RED = auto()
...     GREEN = auto()
...     BLUE = auto()
>>> KeepFlag(2**2 + 2**4)
<KeepFlag.BLUE|16: 20>
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 696)

Unknown directive type "versionadded".

```
.. versionadded:: 3.11
```

Supported `__dunder__` names

`attr: __members__` is a read-only ordered mapping of `member_name` member items. It is only available on the class.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 703); [backlink](#)

Unknown interpreted text role "attr".

`meth: __new__`, if specified, must create and return the enum members; it is also a very good idea to set the member's `attr: __value__` appropriately. Once all the members are created it is no longer used.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 706); [backlink](#)

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 706); [backlink](#)

Unknown interpreted text role "attr".

Supported `__sunder__` names

- `__name__` -- name of the member
- `__value__` -- value of the member; can be set / modified in `__new__`
- `__missing__` -- a lookup function used when a value is not found; may be overridden
- `__ignore__` -- a list of names, either as a `xclass: 'list'` or a `xclass: 'str'`, that will not be transformed into members, and will be removed from the final class

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 719); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 719); [backlink](#)

Unknown interpreted text role "class".

- `_order_` -- used in Python 2/3 code to ensure member order is consistent (class attribute, removed during class creation)
- `_generate_next_value_` -- used to get an appropriate value for an enum member; may be overridden

Note

For standard `class: Enum` classes the next value chosen is the last value seen incremented by one.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 729); [backlink](#)

Unknown interpreted text role "class".

For `class: Flag` classes the next value chosen will be the next highest power-of-two, regardless of the last value seen.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 732); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 735)

Unknown directive type "versionadded".

```
.. versionadded:: 3.6 ``_missing``, ``_order``, ``_generate_next_value``
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 736)

Unknown directive type "versionadded".

```
.. versionadded:: 3.7 ``_ignore``
```

Utilities and Decorators

`auto` can be used in place of a value. If used, the *Enum* machinery will call an *Enum*'s `meth: generate_next_value_` to get an appropriate value. For *Enum* and *IntEnum* that appropriate value will be the last value plus one; for *Flag* and *IntFlag* it will be the first power-of-two greater than the last value; for *StrEnum* it will be the lower-cased version of the member's name.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 745); [backlink](#)

Unknown interpreted text role "meth".

`_generate_next_value_` can be overridden to customize the values used by `auto`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 755)

Unknown directive type "decorator".

```
.. decorator:: property
```

A decorator similar to the built-in `*property*`, but specifically for enumerations. It allows member attributes to have the same names as members themselves.

```
.. note:: the *property* and the member must be defined in separate classes;
for example, the *value* and *name* attributes are defined in the
*Enum* class, and *Enum* subclasses can define members with the
names ``value`` and ``name``.
```

```
.. versionadded:: 3.11
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 768)

Unknown directive type "decorator".

```
.. decorator:: unique
```

A `:keyword:`class`` decorator specifically for enumerations. It searches an enumeration's `:attr:`_members_``, gathering any aliases it finds; if any are found `:exc:`ValueError`` is raised with the details::

```
>>> from enum import Enum, unique
>>> @unique
... class Mistake(Enum):
...     ONE = 1
...     TWO = 2
...     THREE = 3
...     FOUR = 3
...
Traceback (most recent call last):
...
ValueError: duplicate values found in <enum 'Mistake': FOUR -> THREE
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main\Doc\library\enum.rst, line 786)

Unknown directive type "decorator".

```
.. decorator:: verify
```

A `:keyword:`class`` decorator specifically for enumerations. Members from `:class:`EnumCheck`` are used to specify which constraints should be checked on the decorated enumeration.

```
.. versionadded:: 3.11
```

Notes

`class: IntEnum`, `class: StrEnum`, and `class: IntFlag`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 799); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 799); [backlink](#)

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 799); [backlink](#)

Unknown interpreted text role "class".

These three enum types are designed to be drop-in replacements for existing integer- and string-based values; as such, they have extra limitations:

- `__str__` uses the value and not the name of the enum member
- `__format__`, because it uses `__str__`, will also use the value of the enum member instead of its name

If you do not need/want those limitations, you can either create your own base class by mixing in the `int` or `str` type yourself:

```
>>> from enum import Enum
>>> class MyIntEnum(int, Enum):
...     pass
```

or you can reassign the appropriate `meth: 'str'`, etc., in your enum:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\cpython-main) [Doc] [library]enum.rst, line 816); [backlink](#)

Unknown interpreted text role "meth".

```
>>> from enum import IntEnum
>>> class MyIntEnum(IntEnum):
...     __str__ = IntEnum.__str__
```