:mod:`fractions` --- Rational numbers

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

Unknown directive type "module".

```
.. module:: fractions
   :synopsis: Rational numbers.
```

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

Unknown directive type "moduleauthor".

```
.. moduleauthor:: Jeffrey Yasskin <jyasskin at gmail.com>
```

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

Unknown directive type "sectionauthor".

```
.. sectionauthor:: Jeffrey Yasskin <jyasskin at gmail.com>
```

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

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

Unknown interpreted text role "source".

The :mod:`fractions` module provides support for rational number arithmetic.

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

Unknown interpreted text role "mod".

A Fraction instance can be constructed from a pair of integers, from another rational number, or from a string.

The first version requires that *numerator* and *denominator* are instances of :class:`numbers.Rational` and returns a new :class:`Fraction` instance with value `numerator/denominator`. If *denominator* is :const:`0`, it raises a :exc:`ZeroDivisionError`. The second version requires that *other_fraction* is an instance of :class:`numbers.Rational` and returns a :class:`Fraction` instance with the same value. The next two versions accept either a :class:`float` or a :class:`decimal.Decimal` instance, and return a :class:`Fraction` instance with exactly the same value. Note that due to the usual issues with binary floating-point (see :ref:`tut-fp-issues`), the argument to `Fraction(1.1)` is not exactly equal to 11/10, and so `Fraction(1.1)` does *not* return `Fraction(11, 10)` as one might expect. (But see the documentation for the :meth:`limit_denominator` method below.) The last version of the constructor expects a string or unicode instance. The usual form for this instance is:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "const".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "exc".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 26); *backlink*

Unknown interpreted text role "meth".

```
[sign] numerator ['/' denominator]
```

where the optional `sign` may be either '+' or '-' and `numerator` and `denominator` (if present) are strings of decimal digits (underscores may be used to delimit digits as with integral literals in code). In addition, any string that represents a finite value and is accepted by the :class:`float` constructor is also accepted by the :class:`Fraction` constructor. In either form the input string may also have leading and/or trailing whitespace. Here are some examples:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 43); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main] [Doc] [library]fractions.rst, line 43); *backlink*

Unknown interpreted text role "class".

```
>>> from fractions import Fraction
>>> Fraction(16, -10)
```

```
Fraction(-8, 5)
>>> Fraction(123)
Fraction(123, 1)
>>> Fraction()
Fraction(0, 1)
>>> Fraction('3/7')
Fraction(3, 7)
>>> Fraction(' -3/7 ')
Fraction(-3, 7)
>>> Fraction('1.414213 \t\n')
Fraction(1414213, 1000000)
>>> Fraction('-.125')
Fraction(-1, 8)
>>> Fraction('7e-6')
Fraction(7, 1000000)
>>> Fraction(2.25)
Fraction(9, 4)
>>> Fraction(1.1)
Fraction(2476979795053773, 2251799813685248)
>>> from decimal import Decimal
>>> Fraction(Decimal('1.1'))
Fraction(11, 10)
```

The :class:`Fraction` class inherits from the abstract base class :class:`numbers.Rational`, and implements all of the methods and operations from that class. :class:`Fraction` instances are hashable, and should be treated as immutable. In addition, :class:`Fraction` has the following properties and methods:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 78); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 78); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 78); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 78); *backlink*

Unknown interpreted text role "class".

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

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.2
   The :class:`Fraction` constructor now accepts :class:`float` and
   :class:`decimal.Decimal` instances.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 88)

Unknown directive type "versionchanged".

```
.. versionchanged:: 3.9
   The :func:`math.gcd` function is now used to normalize the *numerator*
   and *denominator*. :func:`math.gcd` always return a :class:`int` type.
   Previously, the GCD type depended on *numerator* and *denominator*.
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst`, line 93)

Unknown directive type "versionchanged".

```
   .. versionchanged:: 3.11
      Underscores are now permitted when creating a :class:`Fraction` instance
      from a string, following :PEP:`515` rules.
```

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

Unknown directive type "versionchanged".

```
   .. versionchanged:: 3.11
      :class:`Fraction` implements ``__int__`` now to satisfy
      ``typing.SupportsInt`` instance checks.
```

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

Unknown directive type "attribute".

```
   .. attribute:: numerator

      Numerator of the Fraction in lowest term.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 105)

Unknown directive type "attribute".

```
   .. attribute:: denominator

      Denominator of the Fraction in lowest term.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 110)

Unknown directive type "method".

```
   .. method:: as_integer_ratio()

      Return a tuple of two integers, whose ratio is equal
      to the Fraction and with a positive denominator.

      .. versionadded:: 3.8
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 117)

Unknown directive type "method".

```
   .. method:: from_float(flt)

      This class method constructs a :class:`Fraction` representing the exact
      value of *flt*, which must be a :class:`float`. Beware that
      ``Fraction.from_float(0.3)`` is not the same value as ``Fraction(3, 10)``.

      .. note::

         From Python 3.2 onwards, you can also construct a
         :class:`Fraction` instance directly from a :class:`float`.
```

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

Unknown directive type "method".

```
   .. method:: from_decimal(dec)

      This class method constructs a :class:`Fraction` representing the exact
```

```
   value of *dec*, which must be a :class:`decimal.Decimal` instance.

   .. note::

      From Python 3.2 onwards, you can also construct a
      :class:`Fraction` instance directly from a :class:`decimal.Decimal`
      instance.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 141)

Unknown directive type "method".

```
.. method:: limit_denominator(max_denominator=1000000)

   Finds and returns the closest :class:`Fraction` to ``self`` that has
   denominator at most max_denominator.  This method is useful for finding
   rational approximations to a given floating-point number:

      >>> from fractions import Fraction
      >>> Fraction('3.1415926535897932').limit_denominator(1000)
      Fraction(355, 113)

   or for recovering a rational number that's represented as a float:

      >>> from math import pi, cos
      >>> Fraction(cos(pi/3))
      Fraction(4503599627370497, 9007199254740992)
      >>> Fraction(cos(pi/3)).limit_denominator()
      Fraction(1, 2)
      >>> Fraction(1.1).limit_denominator()
      Fraction(11, 10)
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 162)

Unknown directive type "method".

```
.. method:: __floor__()

   Returns the greatest :class:`int` ``<= self``.  This method can
   also be accessed through the :func:`math.floor` function:

     >>> from math import floor
     >>> floor(Fraction(355, 113))
     3
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 172)

Unknown directive type "method".

```
.. method:: __ceil__()

   Returns the least :class:`int` ``>= self``.  This method can
   also be accessed through the :func:`math.ceil` function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 178)

Unknown directive type "method".

```
.. method:: __round__()
            __round__(ndigits)

   The first version returns the nearest :class:`int` to ``self``,
   rounding half to even. The second version rounds ``self`` to the
   nearest multiple of ``Fraction(1, 10**ndigits)`` (logically, if
   ``ndigits`` is negative), again rounding half toward even.  This
```

```
      method can also be accessed through the :func:`round` function.
```

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\cpython-main\Doc\library\[cpython-main][Doc][library]fractions.rst, line 188)

Unknown directive type "seealso".

```
.. seealso::

   Module :mod:`numbers`
      The abstract base classes making up the numeric tower.
```