
symba
Release 0.6.0

Azat Ibrakov

Jun 28, 2021

CONTENTS

1 base module	3
Python Module Index	7
Index	9

Note: If object is not listed in documentation it should be considered as implementation detail that can change and should not be relied upon.

BASE MODULE

`symba.base.sqrt` (*argument: Union[numbers.Real, symba.core.expression.Expression]*) → `symba.core.expression.Expression`

Returns square root of the argument: exact if it is a perfect square, symbolic instead.

```
>>> sqrt(0) == 0
True
>>> sqrt(1) == 1
True
>>> square_root_of_two = sqrt(2)
>>> square_root_of_two ** 2 == 2
True
>>> 1 < square_root_of_two < 2
True
```

class `symba.base.Expression`

abstract property degree

Returns degree of the expression.

abstract property is_finite

Checks if the expression is finite.

abstract evaluate (*sqr_evaluator: Optional[Callable[[SupportsFloat], numbers.Real]] = None*) → `numbers.Real`

Evaluates the expression.

abstract extract_common_denominator () → `Tuple[int, symba.core.expression.Expression]`

Returns a pair of the common denominator of the expression and the rest of the expression.

abstract extract_common_numerator () → `Tuple[int, symba.core.expression.Expression]`

Returns a pair of the common numerator of the expression and the rest of the expression.

abstract inverse () → `symba.core.expression.Expression`

Returns the expression inverted.

abstract is_positive () → `bool`

Checks if the expression is positive.

abstract lower_bound () → `numbers.Real`

Returns lower bound of the expression.

abstract perfect_sqrt () → `symba.core.expression.Expression`

Returns perfect square root part of the expression.

abstract significant_digits_count () → `int`

Returns significant digits count of the expression.

abstract square () → `symba.core.expression.Expression`
 Returns the expression squared.

abstract upper_bound () → `numbers.Real`
 Returns upper bound of the expression.

__abs__ () → `symba.core.expression.Expression`
 Returns an absolute value of the expression.

abstract __add__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `symba.core.expression.Expression`
 Returns sum of the expression with the other.

__ceil__ () → `int`
 Return the ceiling of the expression.

__floor__ () → `int`
 Return the floor of the expression.

__floordiv__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `int`
 Returns quotient of the division of the expression by the other.

__ge__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `bool`
 Checks if the expression is greater than or equal to the other.

__gt__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `bool`
 Checks if the expression is greater than the other.

abstract __hash__ () → `int`
 Returns hash value of the expression.

__le__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `bool`
 Checks if the expression is lower than or equal to the other.

__lt__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `bool`
 Checks if the expression is lower than the other.

__mod__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `symba.core.expression.Expression`
 Returns remainder of the division of the expression by the other.

abstract __mul__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `symba.core.expression.Expression`
 Returns multiplication of the expression with the other.

abstract __neg__ () → `symba.core.expression.Expression`
 Returns the expression negated.

__pos__ () → `symba.core.expression.Expression`
 Returns the expression positive.

__pow__ (*exponent: int*) → `symba.core.expression.Expression`
 Returns the expression raised to the given exponent.

abstract __radd__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `symba.core.expression.Expression`
 Returns sum of the other with the expression.

__rfloordiv__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `int`
 Returns quotient of the division of the other by the expression.

__rmod__ (*other: Union[`numbers.Real`, `symba.core.expression.Expression`]*) → `symba.core.expression.Expression`
 Returns remainder of the division of the other by the expression.

PYTHON MODULE INDEX

S

`symba.base`, 3

Symbols

__abs__ () (*symba.base.Expression method*), 4
 __add__ () (*symba.base.Expression method*), 4
 __ceil__ () (*symba.base.Expression method*), 4
 __floor__ () (*symba.base.Expression method*), 4
 __floordiv__ () (*symba.base.Expression method*), 4
 __ge__ () (*symba.base.Expression method*), 4
 __gt__ () (*symba.base.Expression method*), 4
 __hash__ () (*symba.base.Expression method*), 4
 __le__ () (*symba.base.Expression method*), 4
 __lt__ () (*symba.base.Expression method*), 4
 __mod__ () (*symba.base.Expression method*), 4
 __mul__ () (*symba.base.Expression method*), 4
 __neg__ () (*symba.base.Expression method*), 4
 __pos__ () (*symba.base.Expression method*), 4
 __pow__ () (*symba.base.Expression method*), 4
 __radd__ () (*symba.base.Expression method*), 4
 __rfloordiv__ () (*symba.base.Expression method*),
 4
 __rmod__ () (*symba.base.Expression method*), 4
 __rmul__ () (*symba.base.Expression method*), 4
 __round__ () (*symba.base.Expression method*), 5
 __rsub__ () (*symba.base.Expression method*), 5
 __rtruediv__ () (*symba.base.Expression method*), 5
 __sub__ () (*symba.base.Expression method*), 5
 __truediv__ () (*symba.base.Expression method*), 5
 __trunc__ () (*symba.base.Expression method*), 5

D

degree () (*symba.base.Expression property*), 3

E

evaluate () (*symba.base.Expression method*), 3
 Expression (*class in symba.base*), 3
 extract_common_denominator ()
 (*symba.base.Expression method*), 3
 extract_common_numerator ()
 (*symba.base.Expression method*), 3

G

get_sqrt_evaluator () (*in module symba.base*), 5

I

inverse () (*symba.base.Expression method*), 3
 is_finite () (*symba.base.Expression property*), 3
 is_positive () (*symba.base.Expression method*), 3

L

lower_bound () (*symba.base.Expression method*), 3

M

module
 symba.base, 3

P

perfect_sqrt () (*symba.base.Expression method*), 3

S

set_sqrt_evaluator () (*in module symba.base*), 5
 significant_digits_count ()
 (*symba.base.Expression method*), 3
 sqrt () (*in module symba.base*), 3
 square () (*symba.base.Expression method*), 3
 symba.base
 module, 3

U

upper_bound () (*symba.base.Expression method*), 4