MikoshiLang

Current Reference Manual

Language, Python API and Function Catalogue

Mikoshi Ltd

Source edition 3.8.0
Updated 4 November 2026

Abstract

MikoshiLang is a Wolfram-style symbolic language running inside Python, built on SymPy and NumPy. This edition documents current features and repaired helpers, with verified runnable examples, all 108 public Python exports, and a catalogue of 4,201 registered library names. Names, constants, aliases and algorithms are different things. MikoshiLang is alpha software: tests support checked behaviours, not correctness for every input, full Wolfram compatibility or equivalence to Mathematica.

Source snapshot: `6d067221be924fe789203ed0fc0a3a8b4a656521`.
Repository: https://github.com/DarrenEdwards111/MikoshiLang
Installation instructions below reproduce this source snapshot; PyPI versions may differ.

Contents

1 What this edition updates **4**
1.1 Companion documentation 4

2 Installation and interfaces **4**
2.1 Session versus direct Python 5

3 Beginner tutorial: your first calculation **5**
3.1 Prepare your environment 5
3.2 Create and run a Python file 5
3.3 Learn the three essential syntax rules 6
3.4 Worked application: a simple cost model 6
3.5 Use the direct Python helpers deliberately 7
3.6 Troubleshooting and next steps 7

4 Language and computation **8**

5 Verification, Python integration and types **8**
5.1 Independent result checks 8
5.2 Python integration 8
5.3 Opt-in type checking 9

6 Corrected helpers and scientific boundaries **9**

7 Test evidence and honest limits **9**

8 Verified runnable examples **11**
8.1 Session: Python data and exact arithmetic 11
8.2 Session: definitions, patterns, and isolation 11
8.3 Session: solution rules, ODEs, and series 12
8.4 Session: verification status and strict type predicates 12
8.5 Session: expose a typed Python function locally 13
8.6 Session: explicit permission for Python calls 13
8.7 Direct Python: List, Take, Drop, and Part 13
8.8 Direct Python: finite Range and callable Table 14
8.9 Direct Python: tiny determinants and complex eigenvalues 15
8.10 Direct Python: units, balancing, and complex convolution 15

9 Public Python API reference **17**
9.1 Expressions, language and Python integration (24 exports) 17
9.2 Algebra, calculus, numerics and constants (15 exports) 20

9.3 Lists, statistics and linear algebra (21 exports) 22
9.4 Elementary functions, number theory and logic (29 exports) 26
9.5 Units, chemistry and signal processing (19 exports) 30

10 Registered library-name catalogue **34**

11 Rebuilding and contributing **56**

1 What this edition updates

This guide replaces the mixed-version manual. Historical domain-pack lists, natural-language examples, hosted-console instructions and retracted function totals are not current capability specifications. Earlier editions remain in Git history. The recent audit repaired existing helpers; adding hundreds of tests did not itself add hundreds of algorithms.

Inventory	Count
Public Python exports	108
Functions / classes / constants	91 / 12 / 5
Registered library names	4,201
Audit-status records	4,884
New public-API audit cases in PR 89	436
Earlier direct-Python regression cases in PR 88	118

The registry is not the entire Wolfram-symbol universe: kernel forms, constants, options and system variables can be counted separately. Registration is not certification of every overload or input. The final catalogue is generated from the live registry, not an old advertised total.

1.1 Companion documentation

- `README.md`: current usage and limitations.
- `docs/library_audit.md`: per-head semantics and repair notes.
- `docs/wolfram_coverage_roadmap.md`: broader symbol coverage.
- `docs/python_interop.md`, `docs/verification.md`, `docs/types.md`.
- `docs/public_python_api_audit.md` and `tests/public_api_audit.json`: bounded per-export evidence.

2 Installation and interfaces

For this exact source edition, use a virtual environment and a source checkout. A published PyPI package may be older.

```
git clone https://github.com/DarrenEdwards111/MikoshiLang.git
cd MikoshiLang
git checkout 6d067221be924fe789203ed0fc0a3a8b4a656521
python -m venv .venv
# Activate .venv using the command for your shell
python -m pip install .
# Optional scientific, plotting and notebook dependencies:
python -m pip install ".[all]"
# Development and property tests:
python -m pip install ".[all,test]" build
```

Core dependencies include SymPy, NumPy and requests. Scientific and plotting operations need the relevant extras. Recent public-API CI exercises Python 3.9, 3.12 and 3.13 on Ubuntu and Windows; macOS was not in that audit matrix.

2.1 Session versus direct Python

```
from mikoshilang import Session
s = Session()
s.evaluate("f[x_] := x^2; f[3]")  # raw result: 9
s.eval("1/3")                     # Fraction(1, 3)
```

`Session.evaluate` returns raw values/expressions; `Session.eval` converts data to Python types. Definitions are session-local. The legacy `evaluate`/`parse_and_eval` interface uses a shared default session.

Imported helpers are a separate interface. Their signatures, result shapes and dispatch paths can differ from similarly named language heads. Direct `Table(fn, spec)` takes a Python callable, not a language string and iterator symbol. Tests of one interface do not automatically test the other.

3 Beginner tutorial: your first calculation

This tutorial starts with Python and introduces MikoshiLang a step at a time. You do not need Wolfram Language experience. Use the source installation above to match this manual; the examples need only the core dependencies.

3.1 Prepare your environment

Install Python and Git first. Open a terminal in the cloned MikoshiLang folder. A virtual environment keeps this project's packages separate from other Python projects. Create it once, then activate it each time you open a new terminal.

Windows PowerShell:

```
py -m venv .venv
.\.venv\Scripts\Activate.ps1
python -m pip install .
```

If PowerShell blocks activation, you do not need to change its policy. Use the environment's Python directly:

```
.\.venv\Scripts\python.exe -m pip install .
.\.venv\Scripts\python.exe hello_mikoshi.py
```

The second command runs the file you will create below.

Linux or macOS terminal:

```
python3 -m venv .venv
source .venv/bin/activate
python -m pip install .
```

On some Linux systems, the distribution's Python venv package must be installed first. These are setup instructions, not a claim that this edition's CI tested macOS.

3.2 Create and run a Python file

Create a plain-text file named `hello_mikoshi.py` in your project folder. Make sure your editor does not add a hidden `.txt` extension. Put this in it:

```
from mikoshilang import Session
```

```
s = Session()
print(s.eval("2 + 3 * 4"))
print(s.eval("1/3 + 1/6"))
print(s.eval("{2, 4, 6}"))
```

With the environment activated, run:

```
python hello_mikoshi.py
```

Expected output:

```
14
1/2
[2, 4, 6]
```

Python runs the outer program. The strings passed to `s.eval` contain MikoshiLang code. A `Session` holds your definitions; `eval` returns convenient Python values. The fraction is exact, rather than a rounded decimal.

3.3 Learn the three essential syntax rules

- Language function calls use square brackets: `square[5]`.
- Language lists use braces: `{2, 4, 6}`. Python output uses square brackets.
- Powers inside the language use `x^2`; ordinary Python uses `x**2`.

Add the following to the same file:

```
s.eval("square[x_] := x^2")
print(s.eval("square[5]"))
print(s.eval("Table[square[k], {k, 1, 4}]"))
```

Expected additional output:

```
25
[1, 4, 9, 16]
```

The pattern `x_` names an argument. The delayed definition `:=` evaluates the right side when the function is called. `Table` repeats a calculation; here `k` runs from 1 through 4, inclusive. Keep using the same session to retain the definition.

3.4 Worked application: a simple cost model

Suppose a small job has a fixed cost of 12 currency units plus 3 per item. Define the cost once and calculate several order sizes:

```
s.eval("cost[n_] := 12 + 3*n")
costs = s.eval("Table[cost[n], {n, 0, 5}]")
print(costs)
print(s.eval("cost[5]"))
assert costs == [12, 15, 18, 21, 24, 27]
```

Expected additional output:

```
[12, 15, 18, 21, 24, 27]
27
```

The first value is the fixed cost at zero items. For five items, a hand check gives $12+3\times5=27$. The assertion is an executable check: Python raises an error if the returned list differs. Try changing the fixed cost or the per-item cost, and update your expected values too.

3.5 Use the direct Python helpers deliberately

The imported Python helpers are a different interface, not interchangeable spellings of session calls:

```
from mikoshilang import List, Take, Range

xs = List(10, 20, 30, 40)
assert Take(xs, -2).args == (30, 40)
assert Range(3, 1, -1).args == (3, 2, 1)
```

Here `List` builds a MikoshiLang expression and `.args` exposes its arguments as a Python tuple. Negative `Take` selects from the end. By contrast, `s.eval("{10, 20}")` returns a normal Python list. Start with `Session.eval` and consult the public API reference before mixing interfaces.

3.6 Troubleshooting and next steps

No module named mikoshilang: Install using the same Python interpreter that runs your file. Check your environment activation, or use the explicit environment interpreter path above.

An expression stays symbolic: Check the function's spelling, capitalisation and supported arguments. A new session will not know definitions made in an earlier one.

Unexpected list or result type: Check whether you used `eval`, `evaluate` or a direct Python helper; they have different return contracts.

An unsupported operation: Read the relevant API entry and limitations. A registered name alone does not promise every possible input form.

Next, work through the verified examples for equation solving, calculus, matrices and units. Use independent expected answers where possible. The assertions in this tutorial were executed against the documented source snapshot; they establish these examples, not correctness for all inputs.

4 Language and computation

Function calls use square brackets and lists use braces. Expressions have heads and arguments; symbols represent variables and function names. Exact integers/rationals differ from machine decimals. Unknown user heads generally remain symbolic; malformed syntax raises an error.

The core includes immediate/delayed definitions, named and typed blank patterns, sequence patterns, conditional rules, replacements, pure functions, list transformations, memoisation and specificity-ordered definitions. `Module`, `Block` and `With` implement documented scoping, including capture-avoiding substitution.

Capabilities include simplification, expansion, factorisation, differentiation, integration, limits, series and supported equation solving. `Solve` returns rule lists, not the bare roots in some old examples. `DSolve` handles supported ODEs and initial conditions, not arbitrary PDEs. `Series` includes the requested order and retains a remainder; `Normal` removes it. Convergence and solver completeness still need problem-specific checking.

`N[expr,p]` can retain arbitrary precision, but requesting more digits cannot recover information lost in machine inputs. Lists, associations, strings, number theory, statistics, matrices, graphs and numerical methods have per-head limits. Consult the catalogue and companion audit rather than assuming every name supports every form.

5 Verification, Python integration and types

5.1 Independent result checks

`Verify[expr]` distinguishes:

Certified An exact certificate was checked with exact arithmetic.

Checked Symbolic or numerical evidence passed; not a general proof.

Unchecked No checker applied or the check was inconclusive.

Failed The result failed its applicable check.

Computation errors can still propagate. Inspect the evidence: expanding a factorisation does not establish irreducibility, and checking returned solutions does not always establish completeness. Methods include primality/product certificates, Bezout identities, substitution, differentiation, numerical comparisons and matrix residuals. This is not a trusted proof-assistant bridge.

5.2 Python integration

`Session.eval` converts rational data to `Fraction` and containers to lists/dictionaries. Request `symbolic="sympy"` or `arrays="numpy"` where appropriate. `Session.ns` provides Pythonic access to language heads; keyword arguments become options.

`@session.function` exposes a typed host function locally. Global `@mikoshilang.function` registration affects all sessions; `unregister` performs cleanup. `from_python`/`to_python` provide explicit conversion. Unsupported objects raise rather than becoming arbitrary foreign atoms.

Python imports/calls are off by default. An explicit `allow_python` allow-list is not an operating-system sandbox. Network knowledge calls need explicit opt-in; a session flag does not sandbox direct Python calls. Untrusted evaluation needs an external process with time and memory limits. Old hosted-service and natural-language promises are not assumed available.

`PackedArray` stores supported machine numbers in NumPy-backed form. Eligible operations use NumPy paths. Unsigned values overflowing int64 are rejected rather than wrapping negative. Packed arrays are not arbitrary-precision storage.

5.3 Opt-in type checking

`Session(check_types=True)` raises on concrete mismatches at checked boundaries; report mode collects mismatches. Symbolic arguments are permitted because their future values are unknown. Types cover integers with sign constraints, rationals, reals, complex numbers, variables, lists, matrices and associations. `TypeSignature`, `TypeMatchQ` and `TypeCheck` are MikoshiLang extensions. See the types guide for supported forms.

6 Corrected helpers and scientific boundaries

Negative direct `Take`/`Drop` counts operate on the tail. `Part[...,0]` returns the list head. Descending ranges/tables respect direction and endpoint; wrong-direction tables are empty without running their body. Zero, nonfinite and nonadvancing table steps error instead of hanging.

Complex eigenvalues/eigenvectors retain imaginary parts. Tiny determinants and matrix entries are not snapped to zero. Integer determinants use exact arithmetic before float conversion could overflow; other direct matrix operations remain numerical. GCD/LCM integer/sign/zero contracts and noninteger Fibonacci validation were repaired. Symbolic logic/comparisons preserve unknown expressions rather than Python truthiness. Calculus and symbolic Fourier helpers retain assumption-bearing SymPy variable identity while calculating; returned Mikoshi symbols do not encode those assumptions.

`Quantity`/`UnitConvert` preserve tiny values and canonical derived SI dimensions but remain floating point. Offset-temperature conversion is supported; ambiguous offset arithmetic is rejected, and temperature differences are not modelled. Element data is not a general isotope database. Neutral molecular formulas and parenthesised groups are supported; malformed formulas and unsupported charge/isotope/hydrate notation error. Chemical balancing uses exact rational arithmetic for unique strictly positive balances, not rounded guesses.

Direct DFT/IDFT use NumPy's negative forward exponent and $1/n$ inverse normalisation; direct symbolic Fourier transforms use SymPy's convention. Do not blindly compare these with the session's Wolfram-style Fourier convention. Complex signal components are preserved. Numerical signal helpers require finite nonempty one-dimensional lists; cutoffs are finite, ordered and strictly below Nyquist. Tests do not guarantee high-order filter stability. SciPy is optional; spectrogram plotting needs matplotlib, otherwise the documented raw frequency/time/power form is returned.

7 Test evidence and honest limits

At the PR 89 audit snapshot, 436 new audit cases plus 118 earlier regressions gave 554 passing focused cases locally, both from source and installed wheels. A minimal environment gave 484 passed and 70 optional skips. Full-suite comparison on Windows/Python 3.14 gave 30,229 passed, 78 pre-existing failures and 9 skips, with zero new failing identities and no tests removed. These are dated results, not guarantees for future revisions.

Pre-merge CI passed the executed source/wheel and minimal jobs on Ubuntu and Windows with Python 3.9, 3.12 and 3.13; optional full-verification jobs were skipped. No licensed Wolfram kernel was used as an oracle. Shared-backend comparisons supplement independent known values, identities and residuals. This public-surface audit is not an exhaustive audit of every registered

head or class method.

8 Verified runnable examples

These examples are checked against the current checkout by `scripts/check_manual_examples.py`. Run it with Python and the core package dependencies installed; no network, credentials, or optional scientific extras are needed. Each listing is a standalone Python program. The output listings are exact stdout checked by the script, including the explicit formatting of numerical results. Exception examples deliberately display only the exception class, not unstable full error messages.

```
python scripts/check_manual_examples.py
python scripts/check_manual_examples.py --write
```

Without `-write`, the script also checks that this generated chapter has not drifted. With `-write`, it regenerates the chapter only after all examples pass. This is a bounded documentation check, not a claim to verify the whole library.

8.1 Session: Python data and exact arithmetic

Use `Session.eval(source)` for Python data. Exact rationals become `fractions.Fraction`; an inexact zero remains a Python float. `Session.evaluate(source)` instead returns the raw language result.

```
from fractions import Fraction
from mikoshilang import Session

s = Session()
assert s.eval("1 + 2") == 3
q = s.eval("1/3")
assert type(q) is Fraction and q == Fraction(1, 3)
z = s.eval("0.5 - 0.5")
assert type(z) is float and z == 0.0
print(s.eval("1 + 2"))
print(repr(q))
print(repr(z))
```

Expected standard output:

```
3
Fraction(1, 3)
0.0
```

8.2 Session: definitions, patterns, and isolation

Definitions and pattern dispatch belong to one session. The integer pattern is more specific than the general pattern, and the literal definition is more specific still.

```
from mikoshilang import Session

s = Session()
s.eval('f[x_] := "general"; f[n_Integer] := n^2; f[0] = 7')
values = s.eval(' {f[0], f[3], f["x"]} ')
assert values == [7, 9, "general"]
assert str(Session().eval("f[3]")) == "f[3]"
print(values)
print(Session().eval("f[3]"))
```

Expected standard output:

```
[7, 9, 'general']
f[3]
```

8.3 Session: solution rules, ODEs, and series

The language heads `Solve` and `DSolve` return nested rule lists, not bare roots or Python dictionaries. Here `evaluate` preserves the raw language braces; `eval` converts the list containers to Python lists while leaving rules as expressions. Other symbolic results remain expressions by default. Series includes the requested order; `Normal` removes the remainder term.

```
from mikoshilang import Session

s = Session()
print(s.evaluate("Solve[{x + y == 3, x - y == 1}, {x, y}]"))
print(s.evaluate("DSolve[{y'[x] == y[x], y[0] == 1}, y[x], x]"))
print(s.eval("Series[Exp[x], {x, 0, 3}]"))
print(s.eval("Normal[Series[Exp[x], {x, 0, 3}]]"))
assert s.eval("{x, y} /. First[Solve[{x+y==3, x-y==1}, {x,y}]]") == [2, 1]
assert s.eval("Normal[Series[Exp[x], {x,0,3}]] /. x -> 0") == 1
```

Expected standard output:

```
{{x -> 2, y -> 1}}
{{y[x] -> E^x}}
1 + x + x^2/2 + x^3/6 + O[x]^4
1 + x + x^2/2 + x^3/6
```

8.4 Session: verification status and strict type predicates

Only the outer status is displayed here, not the full evidence. `Certified` means an exact certificate was checked; `Checked` means symbolic or numerical evidence, not a proof; `Unchecked` means no applicable or conclusive checker. A failed check returns `Failed`; computation errors still propagate. `TypeMatchQ` is a strict predicate: a symbolic value is not known to be an integer. This differs from opt-in call-boundary type checking, which permits symbolic arguments.

```
from mikoshilang import Session

s = Session()
print(s.evaluate("Head[Verify[GCD[12, 18]]]"))
print(s.evaluate("Head[Verify[Integrate[x^2, x]]]"))
print(s.evaluate("Head[Verify[Sort[{3, 1, 2}]]]"))
matches = s.eval("{TypeMatchQ[{1,2}, ListOf[Integer]], TypeMatchQ[x, Integer]}")
assert matches == [True, False]
print(matches)
```

Expected standard output:

```
Certified
Checked
Unchecked
[True, False]
```

8.5 Session: expose a typed Python function locally

The host deliberately exposes a function with `@s.function`; this needs no Python-import permission and does not register a global library head. Type annotations constrain concrete inputs. The error example prints only the checked exception class.

```
from mikoshilang import Session

s = Session()

@s.function(listable=True, check="exact")
def square_plus_one(n: int) -> int:
    return n * n + 1

values = s.eval("SquarePlusOne[{1, 2, 3}]")
assert values == [2, 5, 10]
assert str(Session().eval("SquarePlusOne[2]")) == "SquarePlusOne[2]"
print(values)
try:
    s.eval("SquarePlusOne[2.5]")
except TypeError as exc:
    print(type(exc).__name__)
else:
    raise AssertionError("expected a TypeError")
```

Expected standard output:

```
[2, 5, 10]
TypeError
```

8.6 Session: explicit permission for Python calls

Python imports/calls are disabled by default. An allow-list enables the named modules and their submodules; it is not a security sandbox. Do not execute untrusted input without an external process sandbox. This offline example uses only the standard library.

```
from mikoshilang import Session

call = 'PythonFunction["math.gcd"][12, 18]'
try:
    Session().eval(call)
except PermissionError as exc:
    print(type(exc).__name__)
else:
    raise AssertionError("Python calls must be gated")
s = Session(allow_python={"math"})
assert s.eval(call) == 6
print(s.eval(call))
```

Expected standard output:

```
PermissionError
6
```

8.7 Direct Python: List, Take, Drop, and Part

These imported helpers are not `Session` language calls. `Take(lst, n)`, `Drop(lst, n)`, and `Part(lst, *indices)` accept a `List` expression, not an ordinary Python list. Negative counts

select/drop from the end; zero counts select nothing/drop nothing. Part indices are one-based, negative indices count from the end, and index zero returns the head.

```
from mikoshilang import List, Take, Drop, Part

xs = List(10, 20, 30, 40)
assert Take(xs, -2).args == (30, 40)
assert Drop(xs, -2).args == (10, 20)
assert Take(xs, 0).args == ()
assert Drop(xs, 0) == xs
assert Part(xs, -1) == 40
print(Take(xs, -2))
print(Drop(xs, -2))
print(Take(xs, 0))
print(Drop(xs, 0))
print(Part(xs, 0))
print(Part(xs, -1))
try:
    Take([10, 20], 1)
except TypeError as exc:
    print(type(exc).__name__)
else:
    raise AssertionError("expected a List expression requirement")
```

Expected standard output:

```
{30, 40}
{10, 20}
{}
{10, 20, 30, 40}
List
40
TypeError
```

8.8 Direct Python: finite Range and callable Table

`Range(n)`, `Range(a, b)`, and `Range(a, b, step)` use integer bounds with an inclusive end. Direct `Table(fn, spec)` takes a Python callable and either a count, `(min, max)`, or `(min, max, step)`; it does not take a language expression and iterator symbol. Wrong-direction steps return an empty list rather than looping. Table rejects zero, nonfinite, and nonadvancing steps. The verifier runs this entire example in a timeout-limited child process; the checks below include wrong-direction and nonadvancing cases.

```
from mikoshilang import Range, Table

assert Range(3, 1, -1).args == (3, 2, 1)
assert Range(1, 3, -1).args == ()
assert Table(lambda i: i*i, (3, 1, -1)).args == (9, 4, 1)
assert Table(lambda i: i, (1, 3, -1)).args == ()
print(Range(3, 1, -1))
print(Range(1, 3, -1))
print(Table(lambda i: i*i, (3, 1, -1)))
print(Table(lambda i: i, (1, 3, -1)))
for spec in [(1, 3, 0), (1, 3, float("inf")),
             (1e20, 1e20 + 1e6, 1.0)]:
    try:
        Table(lambda i: i, spec)
    except ValueError as exc:
        print(type(exc).__name__)
    else:
        raise AssertionError("expected invalid-step rejection")
```

Expected standard output:

```
{3, 2, 1}
{}
{9, 4, 1}
{}
ValueError
ValueError
ValueError
```

8.9 Direct Python: tiny determinants and complex eigenvalues

Direct linear-algebra helpers use `Matrix(*rows)`, whereas Session language calls use nested lists. Integer determinant input is exact; floating input is numerical and a tiny nonzero determinant must not be rounded to zero. Complex eigenvalues keep their imaginary parts. Formatting below normalizes insignificant numerical variation; the assertions check values, not just display or shape.

```
from math import isclose
from mikoshilang import Matrix, Det, Eigenvalues

assert Det(Matrix([2, 0], [0, 3])) == 6
tiny = Det(Matrix([1e-12, 0.0], [0.0, 1.0]))
assert tiny != 0 and isclose(tiny, 1e-12, rel_tol=1e-12, abs_tol=0)
values = Eigenvalues(Matrix([1j, 0], [0, -1j])).args
assert len(values) == 2 and set(values) == {-1j, 1j}
print(f"{tiny:.0e}")
print([complex(v) == -1j or complex(v) == 1j for v in values])
print([v.imag for v in values])
```

Expected standard output:

```
1e-12
[True, True]
[-1.0, 1.0]
```

8.10 Direct Python: units, balancing, and complex convolution

`UnitConvert(Quantity(value, unit), target)` returns a Python Quantity object. Offset-temperature conversion is supported, but offset-temperature arithmetic is rejected: convert to kelvin first. `BalanceEquation(equation)` accepts an equation string; `Convolve(a, b)` accepts List expressions and preserves complex components. These examples require no SciPy extra.

```
from mikoshilang import (
    Quantity, UnitConvert, BalanceEquation, Convolve, List,
)

distance = UnitConvert(Quantity(1, "km"), "m")
assert distance.value == 1000.0 and distance.unit_name == "m"
temperature = UnitConvert(Quantity(0, "C"), "K")
assert temperature.value == 273.15 and temperature.unit_name == "K"
print(distance)
print(temperature)
try:
    Quantity(0, "C") + Quantity(1, "C")
except ValueError as exc:
    print(type(exc).__name__)
else:
    raise AssertionError("offset-temperature arithmetic must be rejected")
balanced = BalanceEquation("H2 + O2 -> H2O")
```

```
assert balanced == "2H2 + O2 -> 2H2O"
print(balanced)
result = Convolve(List(1j, 1), List(1, 1j))
assert result.args == (1j, 0, 1j)
print(result.args)
```

Expected standard output:

```
Quantity[1000, "m"]
Quantity[273.15, "K"]
ValueError
2H2 + O2 -> 2H2O
(1j, 0, 1j)
```

9 Public Python API reference

This section lists all 108 names exported by `mikoshilang.__all__`. Signatures and first-line descriptions come from the live Python objects. Each entry records a bounded-audit limitation, not a guarantee of complete method or domain coverage.

9.1 Expressions, language and Python integration (24 exports)

Expr (class)

```
Expr(head: 'str', *args: 'Any')
```

A composite expression with a head and arguments. *Audit scope:* Structural equality/hash and selected arithmetic overloads; arbitrary invalid constructor arguments and every operator combination are not exhaustively checked.

Symbol (class)

```
Symbol(name: 'str')
```

A symbolic variable like x, y, Pi. *Audit scope:* Preserves documented structural comparisons; identifier validation and every overloaded operator are not exhaustively checked.

symbols (function)

```
symbols(*names: 'str')
```

Create multiple symbols: x, y, z = symbols('x', 'y', 'z') *Audit scope:* Single/multiple names and whitespace splitting checked; no new contract imposed for zero names.

Blank (class)

```
Blank(name: 'Optional[str]' = None, head: 'Optional[str]' = None)
```

Matches any single expression. Optionally binds a name and/or constrains head type. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

BlankSequence (class)

```
BlankSequence(name: 'Optional[str]' = None)
```

Matches one or more expressions. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

BlankNullSequence (class)

```
BlankNullSequence(name: 'Optional[str]' = None)
```

Matches zero or more expressions. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

Pattern (class)

```
Pattern(head: 'str', *args)
```

A pattern expression — an Expr-like structure with Blank slots. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

Rule (class)

```
Rule(pattern, replacement)
```

A substitution rule: pattern -> replacement. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

RuleDelayed (class)

```
RuleDelayed(pattern, replacement_fn: 'Callable[[Dict[str, Any]], Any]')
```

A lazy substitution rule: pattern :> replacement_fn. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

replace (function)

```
replace(expr, rules) -> 'Any'
```

Apply first matching rule to expr (top level only). *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

replace_all (function)

```
replace_all(expr, rules, _depth=0) -> 'Any'
```

Apply rules recursively throughout the expression. *Audit scope:* Existing repeated-rewrite semantics and depth cap retained, not changed to Wolfram single-pass semantics. Cyclic/growing rewrites bounded in 20-second subprocess.

evaluate (function)

```
evaluate(expr, depth: 'int' = 0) -> 'Any'
```

Evaluate an expression with Wolfram-style semantics. *Audit scope:* Direct exported tree evaluator tested; source strings remain literal values. Full kernel/library scope intentionally excluded.

parse (function)

```
parse(text: 'str') -> 'Any'
```

Parse a single Wolfram-style expression string into a MikoshiLang Expr. *Audit scope:* Representative grammar/precedence and malformed-input checks; not a complete Wolfram-language grammar audit.

parse_file (function)

```
parse_file(path: 'str') -> 'ListType[Any]'
```

Parse a .ml file, returning a list of expressions (one per line/statement). *Audit scope:* Current multi-statement CompoundExpression structure retained despite one-per-statement docstring; no speculative return-shape change. UTF-8/non-ASCII file decoding not checked.

parse_and_eval (function)

```
parse_and_eval(text: 'str') -> 'Any'
```

Parse a Wolfram-style expression and evaluate it. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

ParseError (class)

```
ParseError: inspect the constructor or source documentation
```

Audit scope: Representative direct public value and edge assertions; not exhaustive over all expression trees.

to_latex (function)

```
to_latex(expr, parent_prec: 'int' = 0) -> 'str'
```

Convert a MikoshiLang expression to a LaTeX string. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

Session (class)

```
Session(allow_network=False, allow_python=False, check_types=False)
```

Independent definitions for one notebook/user; not an execution sandbox. *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

clear_definitions (function)

```
clear_definitions()
```

Audit scope: Default-state mutation runs in a subprocess to avoid clearing unrelated suite state; local Session independence verified.

function (function)

```
function(fn: 'Optional[Callable]' = None, *, name: 'Optional[str]' = None, area: 'str' = 'user
    library', listable: 'bool' = False, notes: 'str' = '', check: 'str' = 'delegated', override: '
    bool' = False)
```

Decorator for library authors: register a typed Python function as a head in every session. *Audit scope:* Global decorator checked with exact inputs, type/value failures, listability and finally-unregister; no edits to evaluator/kernel/library.

unregister (function)

```
unregister(name: 'str') -> 'None'
```

Remove a head registered with ``library_function`` (restoring a library head it overrode). *Audit scope:* Representative direct public value and edge assertions; not exhaustive over all expression trees.

from_python (function)

```
from_python(obj: 'Any') -> 'Any'
```

Convert a Python value to a MikoshiLang expression (see the module table); unknown types raise TypeError. *Audit scope:* Bounded properties (60 nested values, 60 complex pairs), not exhaustive foreign-type/array coverage; Hypothesis tests individually skip when dependency absent.

to_python (function)

```
to_python(e: 'Any', symbolic: 'str' = 'expr', arrays: 'str' = 'list') -> 'Any'
```

Convert a MikoshiLang value to Python. *Audit scope:* Bounded documented-data roundtrips and symbolic/numpy modes; not exhaustive foreign-type coverage.

PackedArray (class)

```
PackedArray(array)
```

A List of machine numbers stored as one read-only ndarray (int64, float64 or complex128). *Audit scope:* Machine integer/float/complex values, dtype conversion, read-only storage and copies checked; broader fast-operation correctness owned by other batches.

9.2 Algebra, calculus, numerics and constants (15 exports)

simplify (function)

```
simplify(expr) -> 'Any'
```

Algebraic simplification. *Audit scope:* Bounded identities, complex values, branch safety and 401-digit exact integers; not exhaustive simplification or domain equivalence coverage.

expand (function)

```
expand(expr) -> 'Any'
```

Expand products and powers. *Audit scope:* Polynomial degrees bounded to three; 30 deterministic Hypothesis examples with small integer coefficients and evaluation points.

factor (function)

```
factor(expr) -> 'Any'
```

Factor an expression. *Audit scope:* Value preservation checked for bounded polynomials over integer, rational and complex evaluation points; no optimal-factorization claim.

diff (function)

```
diff(expr, var, n: 'int' = 1) -> 'Any'
```

Differentiate expr with respect to var, n times. *Audit scope:* Input SymPy symbols retain their identity during calculation; returned Mikoshi Symbol does not encode SymPy assumptions.

integrate (function)

```
integrate(expr, var, *limits) -> 'Any'
```

Integrate expr with respect to var. Optional limits for definite integrals. *Audit scope:* Polynomial and one elementary improper integral tested, not arbitrary convergence/domain analysis.

solve (function)

```
solve(expr, var) -> 'Any'
```

Solve expr == 0 for var. *Audit scope:* Python helper returns List of roots, whereas Session Solve returns rules; intentionally not asserted equal.

limit (function)

```
limit(expr, var, point) -> 'Any'
```

Compute limit of expr as var -> point. *Audit scope:* Direct helper inherits backend right-hand default at finite points; limit(1/x,x,0)==Infinity is not a two-sided-limit guarantee.

series (function)

```
series(expr, var, point=0, order=6) -> 'Any'
```

Taylor polynomial through order ``order`` inclusive (Wolfram's Normal[Series[...]]). *Audit scope:* Python series includes requested order and removes O terms, unlike Session Series; parity checked only with Normal[Series[...]].

N (function)

```
N(expr, precision: 'int' = 15) -> 'Any'
```

Evaluate expression numerically to given precision. *Audit scope:* Exact complex and Decimal sqrt(2)/rational references checked through 60 requested digits; high precision retains SymPy result types.

numerical (function)

```
numerical(expr, precision: 'int' = 15) -> 'Any'
```

Evaluate expression to floating point with given precision. *Audit scope:* Tiny 1e-100 result uses abs=0 tolerance; no arbitrary underflow/overflow guarantee.

Pi (constant)

Symbolic constant: `Pi`. *Audit scope:* Symbol identity, machine value and known high-precision decimal prefix tested; not arbitrary-precision proof.

E (constant)

Symbolic constant: `E`. *Audit scope:* Symbol identity, math.e machine value and exact Log[E] identity tested.

I (constant)

Symbolic constant: `I`. *Audit scope:* Exact square, Euler identity, machine value and one high-precision complex rational tested; no exhaustive complex branch semantics.

Infinity (constant)

Symbolic constant: `Infinity`. *Audit scope:* Positive/negative numerical infinity, high-precision infinity and elementary improper integral/limit checked; indeterminate infinity arithmetic not exhaustively covered.

GoldenRatio (constant)

Symbolic constant: `GoldenRatio`. *Audit scope:* Symbol identity, independent (1+sqrt(5))/2 machine value and exact quadratic identity tested.

9.3 Lists, statistics and linear algebra (21 exports)

List (function)

```
List(*args) -> 'Expr'
```

Create an ordered list. *Audit scope:* Structural constructor; no automatic evaluation or conversion of nested Python lists.

Table (function)

```
Table(expr_fn: 'Callable', spec) -> 'Expr'
```

Generate a list. spec = (min, max) or (min, max, step). *Audit scope:* Python callable with scalar count or two/three-item numeric spec, not symbolic Wolfram iterator syntax. Resource-unbounded output is outside this audit.

Map (function)

```
Map(fn: 'Callable', lst: 'Expr') -> 'Expr'
```

Apply fn to each element of a List. *Audit scope:* Python callable; operates on List Expr only, not raw Python lists.

Select (function)

```
Select(lst: 'Expr', cond: 'Callable') -> 'Expr'
```

Filter elements of a List by condition. *Audit scope:* Python callable predicate and Python truthiness; no symbolic predicate evaluation.

Sort (function)

```
Sort(lst: 'Expr', key=None) -> 'Expr'
```

Sort a List. *Audit scope:* Python ordering and key callable; mutually incomparable mixed values raise TypeError rather than Wolfram canonical ordering.

Reverse (function)

```
Reverse(lst: 'Expr') -> 'Expr'
```

Reverse a List. *Audit scope:* Checked cases are documented in the public-API audit manifest.

Take (function)

```
Take(lst: 'Expr', n: 'int') -> 'Expr'
```

Take n elements from the front, or the last -n when n is negative. *Audit scope:* Python scalar integer count, slicing clamps oversize counts; no multidimensional/spec-list overloads.

Drop (function)

```
Drop(lst: 'Expr', n: 'int') -> 'Expr'
```

Drop n elements from the front, or the last -n when n is negative. *Audit scope:* Python scalar integer count, slicing clamps oversize counts; no multidimensional/spec-list overloads.

Part (function)

```
Part(lst: 'Expr', *indices: 'int') -> 'Any'
```

Access 1-based elements; negative indices count from the end, 0 gives the head. *Audit scope:* Only List Expr traversal and scalar integer indices; no arbitrary expression or list-of-indices overload.

Range (function)

```
Range(*args) -> 'Expr'
```

Range(n), Range(a, b), or Range(a, b, step). *Audit scope:* Python integer range contract; rational/float steps unsupported. Bounded ranges only in audit.

Total (function)

```
Total(lst: 'Expr') -> 'Any'
```

Sum of elements. *Audit scope:* Uses Python sum; symbolic results remain arithmetic expressions rather than automatically simplified Session results.

Mean (function)

```
Mean(lst: 'Expr') -> 'float'
```

Arithmetic mean. *Audit scope:* Python statistics.mean adapter, no symbolic or matrix-column statistics; empty input raises StatisticsError. Fractions stay exact.

Median (function)

```
Median(lst: 'Expr') -> 'float'
```

Median value. *Audit scope:* Python statistics.median ordering contract, not symbolic canonical ordering; empty input raises StatisticsError. Fractions stay exact.

StandardDeviation (function)

```
StandardDeviation(lst: 'Expr') -> 'float'
```

Standard deviation (sample). *Audit scope:* Python statistics.stdev sample convention (n-1); fewer than two values raise StatisticsError; rational input produces numerical square root, not symbolic radical.

Matrix (function)

```
Matrix(*rows) -> 'Expr'
```

Create a matrix from row lists. *Audit scope:* Structural row constructor; accepts empty/ragged/symbolic rows without validating numerical suitability. Consumers perform numeric/shape validation.

Dot (function)

```
Dot(a: 'Expr', b: 'Expr') -> 'Expr'
```

Matrix multiplication. *Audit scope:* Two Matrix expressions, not Wolfram vector/general tensor overloads. NumPy floating/complex backend, including coercion of rationals; no arbitrary-precision or symbolic multiplication contract.

Det (function)

```
Det(m: 'Expr') -> 'int | float | complex'
```

Matrix determinant; integer input is exact, other input is numerical. *Audit scope:* Integer matrices use exact determinant; other accepted numerics use NumPy floating/complex computation. Symbolic entries unsupported; empty/rectangular matrices rejected.

Inverse (function)

```
Inverse(m: 'Expr') -> 'Expr'
```

Matrix inverse. *Audit scope:* Numerical square Matrix inverse; singular/empty/rectangular matrices raise LinAlgError. No exact rational or symbolic inverse contract.

Eigenvalues (function)

```
Eigenvalues(m: 'Expr') -> 'Expr'
```

Eigenvalues of a matrix. *Audit scope:* Numerical square Matrix spectrum; sorted real/imaginary values. Eigenvectors order is not assumed aligned. No symbolic eigenvalue contract.

Eigenvectors (function)

```
Eigenvectors(m: 'Expr') -> 'Expr'
```

Eigenvectors of a matrix. *Audit scope:* Numerical normalized vectors; phases/order are backend-dependent; defective matrices need not yield an independent basis. Tests assert residuals, not arbitrary phases.

Transpose (function)

```
Transpose(m: 'Expr') -> 'Expr'
```

Matrix transpose. *Audit scope:* Numerical Matrix backend coerces values to float/complex; symbolic/huge exact arithmetic not supported. Unshaped Matrix() has no guaranteed matrix-shape preservation.

9.4 Elementary functions, number theory and logic (29 exports)

Sin (function)

```
Sin(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Cos (function)

```
Cos(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Tan (function)

```
Tan(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

ArcSin (function)

```
ArcSin(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

ArcCos (function)

```
ArcCos(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

ArcTan (function)

```
ArcTan(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Exp (function)

```
Exp(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Log (function)

```
Log(x, base=None) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Log2 (function)

```
Log2(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Log10 (function)

```
Log10(x) -> 'Expr'
```

Audit scope: Checked cases are documented in the public-API audit manifest.

Prime (function)

```
Prime(n: 'int') -> 'int'
```

Return the nth prime number. *Audit scope:* Prime property indices are bounded to 1..25; no unbounded prime searches.

PrimeQ (function)

```
PrimeQ(n: 'int') -> 'bool'
```

Test if n is prime. *Audit scope:* Checked cases are documented in the public-API audit manifest.

FactorInteger (function)

```
FactorInteger(n: 'int') -> 'Expr'
```

Factor an integer into prime factors. *Audit scope:* Checked cases are documented in the public-API audit manifest.

GCD (function)

```
GCD(*args: 'int') -> 'int'
```

Greatest common divisor. *Audit scope:* Checked cases are documented in the public-API audit manifest.

LCM (function)

```
LCM(*args: 'int') -> 'int'
```

Least common multiple. *Audit scope:* Checked cases are documented in the public-API audit manifest.

Mod (function)

```
Mod(a: 'int', b: 'int') -> 'int'
```

Modular arithmetic. *Audit scope:* Retains existing Python remainder behavior: native Mikoshi Symbol rejected; SymPy integer symbols supported. No speculative symbolic extension.

Factorial (function)

```
Factorial(n: 'int') -> 'int'
```

n! *Audit scope:* Checked cases are documented in the public-API audit manifest.

Binomial (function)

```
Binomial(n: 'int', k: 'int') -> 'int'
```

Binomial coefficient C(n, k). *Audit scope:* Checked cases are documented in the public-API audit manifest.

Fibonacci (function)

```
Fibonacci(n: 'int') -> 'int'
```

nth Fibonacci number. *Audit scope:* Checked cases are documented in the public-API audit manifest.

And (function)

```
And(*args: 'bool')
```

Audit scope: Direct symbolic calls construct expressions; simplification is tested through Session rather than assumed eager.

Or (function)

```
Or(*args: 'bool')
```

Audit scope: Direct symbolic calls construct expressions; simplification is tested through Session rather than assumed eager.

Not (function)

```
Not(x: 'bool')
```

Audit scope: Direct symbolic calls construct expressions; simplification is tested through Session rather than assumed eager.

Xor (function)

```
Xor(a: 'bool', b: 'bool')
```

Audit scope: Direct symbolic calls construct expressions; simplification is tested through Session rather than assumed eager.

Implies (function)

```
Implies(a: 'bool', b: 'bool')
```

Audit scope: Direct symbolic calls construct expressions; simplification is tested through Session rather than assumed eager.

Equal (function)

```
Equal(a, b)
```

Audit scope: Unresolved symbolic relations remain expression trees; numeric comparisons keep Python behavior, including TypeError on complex ordering.

Less (function)

```
Less(a, b)
```

Audit scope: Unresolved symbolic relations remain expression trees; numeric comparisons keep Python behavior, including TypeError on complex ordering.

Greater (function)

```
Greater(a, b)
```

Audit scope: Unresolved symbolic relations remain expression trees; numeric comparisons keep Python behavior, including TypeError on complex ordering.

LessEqual (function)

```
LessEqual(a, b)
```

Audit scope: Unresolved symbolic relations remain expression trees; numeric comparisons keep Python behavior, including TypeError on complex ordering.

GreaterEqual (function)

```
GreaterEqual(a, b)
```

Audit scope: Unresolved symbolic relations remain expression trees; numeric comparisons keep Python behavior, including TypeError on complex ordering.

9.5 Units, chemistry and signal processing (19 exports)

Quantity (class)

```
Quantity(value: 'float', unit_name: 'str')
```

A finite real float with a named unit or canonical SI dimensions. *Audit scope:* Finite real binary floats; no exact or complex quantities. Offset-temperature arithmetic deliberately errors; convert to K first. Unit grammar is named units plus canonical SI products, not a general parser. Equality retains existing absolute tolerance.

UnitConvert (function)

```
UnitConvert(qty: 'Quantity', target_unit: 'str') -> 'Quantity'
```

Convert a Quantity to a different unit. *Audit scope:* Finite floating-point conversion; no temperature-difference type or physical absolute-zero validation.

Element (function)

```
Element(key) -> 'Expr'
```

Get element data as an Expr. *Audit scope:* Embedded 118-element table; audit checks selected known records, not independent verification of every scientific datum.

AtomicMass (function)

```
AtomicMass(key) -> 'float'
```

Get atomic mass of an element. *Audit scope:* Embedded rounded atomic masses, not isotope-specific mass data.

ElectronConfiguration (function)

```
ElectronConfiguration(key) -> 'str'
```

Get electron configuration of an element. *Audit scope:* Embedded configurations; selected known entries checked, not independent validation of every configuration.

MolecularMass (function)

```
MolecularMass(formula: 'str') -> 'float'
```

Mass rounded to 0.001 for neutral formulas and parenthesized groups. *Audit scope:* Neutral formulas and parenthesized groups only; no hydrates, charges, isotope notation or leading coefficients; results round to 0.001.

BalanceEquation (function)

```
BalanceEquation(equation: 'str') -> 'str'
```

Balance neutral formulas with unique strictly positive coefficients. *Audit scope:* Unique strictly positive neutral-formula balances only; ambiguous, impossible, zero-coefficient or pre-coefficient input equations error. No ionic charge conservation.

FourierTransform (function)

```
FourierTransform(expr, t_var, w_var) -> 'Any'
```

Symbolic Fourier transform using SymPy. *Audit scope:* SymPy Fourier convention; bounded Gaussian/zero cases do not establish convergence for arbitrary symbolic input. Returned Mikoshi symbols do not encode SymPy assumptions.

InverseFourierTransform (function)

```
InverseFourierTransform(expr, w_var, t_var) -> 'Any'
```

Symbolic inverse Fourier transform using SymPy. *Audit scope:* SymPy inverse convention; arbitrary symbolic transforms may remain unevaluated. Returned Mikoshi symbols do not encode SymPy assumptions.

DFT (function)

```
DFT(data: 'Expr') -> 'Expr'
```

NumPy FFT: negative exponent and no forward normalization. *Audit scope:* Direct NumPy FFT convention, not Wolfram Session convention. Finite one-dimensional Expr List inputs; bounded samples only.

IDFT (function)

```
IDFT(data: 'Expr') -> 'Expr'
```

NumPy inverse FFT: positive exponent with 1/n normalization. *Audit scope:* Direct NumPy inverse FFT convention; binary floating-point error is expected.

LowPassFilter (function)

```
LowPassFilter(data: 'Expr', cutoff: 'float', sample_rate: 'float' = 1.0, order: 'int' = 5) -> '
    Expr'
```

Apply a Butterworth low-pass filter. *Audit scope:* Optional SciPy Butterworth/filtfilt direct-form implementation. Positive integer order and finite positive sample rate required; short inputs and invalid cutoffs error. High-order numerical stability and broad frequency-response accuracy not established.

HighPassFilter (function)

```
HighPassFilter(data: 'Expr', cutoff: 'float', sample_rate: 'float' = 1.0, order: 'int' = 5) -> '
    Expr'
```

Apply a Butterworth high-pass filter. *Audit scope:* Optional SciPy Butterworth/filtfilt direct-form implementation. Positive integer order and finite positive sample rate required; short inputs and invalid cutoffs error. High-order numerical stability and broad frequency-response accuracy not established.

BandPassFilter (function)

```
BandPassFilter(data: 'Expr', low: 'float', high: 'float', sample_rate: 'float' = 1.0, order: 'int'
     = 5) -> 'Expr'
```

Apply a Butterworth band-pass filter. *Audit scope:* Optional SciPy Butterworth/filtfilt direct-form implementation. Positive integer order and finite positive sample rate required; short inputs and invalid cutoffs error. High-order numerical stability and broad frequency-response accuracy not established.

Convolve (function)

```
Convolve(signal1: 'Expr', signal2: 'Expr') -> 'Expr'
```

Convolve two signals. *Audit scope:* Full discrete one-dimensional convolution of finite nonempty Expr Lists; no alternate modes.

HammingWindow (function)

```
HammingWindow(n: 'int') -> 'Expr'
```

Generate a Hamming window of length n. *Audit scope:* NumPy symmetric window; nonnegative integer length (including empty and singleton). Bounded known-value checks only.

HanningWindow (function)

```
HanningWindow(n: 'int') -> 'Expr'
```

Generate a Hanning window of length n. *Audit scope:* NumPy symmetric window; nonnegative integer length (including empty and singleton). Bounded known-value checks only.

BlackmanWindow (function)

```
BlackmanWindow(n: 'int') -> 'Expr'
```

Generate a Blackman window of length n. *Audit scope:* NumPy symmetric window; nonnegative integer length (including empty and singleton). Bounded known-value checks only.

Spectrogram (function)

```
Spectrogram(data: 'Expr', sample_rate: 'float' = 1.0) -> 'Any'
```

Compute a PSD spectrogram (two-sided for complex signals). *Audit scope:* Requires SciPy; matplotlib figure path tested when installed, raw fallback forced separately. Uses SciPy defaults; broad PSD accuracy and short-record estimator quality not established.

10 Registered library-name catalogue

The live registry contains 4,201 unique names. This alphabetical list is complete for that registry snapshot, but excludes kernel-only forms, options and constants that are not registry entries. It includes aliases and registered operations with restricted or unsupported forms: registration does not imply that every input works. Consult `docs/library_audit.md` for per-head notes and `docs/wolfram_coverage_roadmap.md` for the broader symbol universe.

$

$CurrentTask
$RandomGeneratorState

A

AASTriangle
ANOVA
APRtoEAR
ARCHProcess
ARIMAProcess
ARMAProcess
ARProcess
ASATriangle
AbelianGroup
Abort
AbortKernels
AbortProtect
Abs
AbsArg
AbsArgPlot
AbsoluteCorrelation
AbsoluteCorrelationFunction
AbsoluteDashing
AbsoluteFileName
AbsoluteOptions
AbsolutePointSize
AbsoluteThickness
AbsoluteTime
AbsoluteTiming
Acceleration
AccountingForm
Accumulate
Accuracy
Ackermann
AcousticAbsorbingValue
AcousticImpedanceValue
AcousticNormalVelocityValue
AcousticPDEComponent
AcousticPressureCondition
AcousticRadiationValue
AcousticSoundHardValue
AcousticSoundSoftCondition
Activate
AcyclicGraphQ
AddSides
AddToSearchIndex
AdjacencyGraph
AdjacencyList
AdjacencyMatrix
AdjacentMeshCells
Adjugate
AdjustTimeSeriesForecast
AffineHalfSpace
AffineSpace
AffineTransform
AggregateRows
AiryAi
AiryAiPrime
AiryAiZero
AiryBi
AiryBiPrime
AiryBiZero
AlgebraicIntegerQ
AlgebraicNumber
AlgebraicNumberDenominator
AlgebraicNumberNorm
AlgebraicNumberPolynomial
AlgebraicNumberTrace
AlgebraicUnitQ
Algebraics
AllMatch
AllSAT
AllSameBy
AllTrue
AlphaChannel
Alphabet
AlphabetPosition
AlphabeticOrder
AlphabeticSort
AlternatingFactorial
AlternatingGroup
AlternatingHarmonicNumber
AlternativeHypothesis
AmbientLight
AndersonDarling
AndersonDarlingTest
AngerJ
AngleBisector
AngleBracket
AnglePath
AnglePath3D
AngleVector
AngularGauge
AnimatePlot
Annotate
Annotation
AnnotationDelete
AnnotationKeys
AnnotationValue
AnnuityFV
AnnuityPV
Annulus
Antialiasing
Anticommutator
Antihermitian
AntihermitianMatrixQ
Antisymmetric
AntisymmetricMatrixQ
AnyMatch
AnySubset
AnyTrue
ApartSquareFree
AppellF1
AppellF2
AppellF3
AppellF4
AppendTo
ApplySides
ArcCos
ArcCosh
ArcCot
ArcCoth
ArcCsc
ArcCsch
ArcCurvature
ArcLength
ArcSec
ArcSech
ArcSin
ArcSinDistribution
ArcSinh
ArcTan
ArcTanh
Area
Arg
ArgMax
ArgMin
ArithmeticGeometricMean
ArithmeticSequence
Around
AroundReplace
Array
ArrayComponents
ArrayDepth
ArrayDot
ArrayExpand
ArrayFilter
ArrayFlatten
ArrayMesh
ArrayPad
ArrayPlot
ArrayPlot3D
ArrayQ
ArrayReduce
ArrayResample
ArrayReshape
ArrayRules
ArraySimplify
ArraySymbol
Arrays
Arrow
Arrowheads
AspectRatio
Assert
AssociateTo
Association
AssociationComap
AssociationMap
AssociationQ
AssociationThread
Assuming

Assumptions
Asymptotic
AsymptoticDSolveValue
AsymptoticEqual
AsymptoticEquivalent
AsymptoticExpectation
AsymptoticGreater
AsymptoticGreaterEqual
AsymptoticIntegrate
AsymptoticLess
AsymptoticLessEqual
AsymptoticProbability
AsymptoticProduct
AsymptoticRSolveValue
AsymptoticSolve
AsymptoticSum
AtomicMassUnit
AugmentedPolyhedron
AugmentedSymmetricPolynomial
AutocorrelationTest
AvogadroNumber
Axes
AxesEdge
AxesLabel
AxesOrigin
AxesStyle
Axis
AxisLabel
AxisObject
AxisStyle

B

BSplineBasis
BSplineCurve
BSplineFunction
BSplineSurface
BabyMonsterGroupB
Backslash
Ball
Band
BandpassFilter
BandstopFilter
BarChart
BarChart3D
BarLegend
BarOrigin
BarSpacing
BarabasiAlbertGraphDistribution
BarcodeImage
BaringhausHenzeTest
BarnesG
Base64Decode
Base64Encode
BaseConvert
BaseDecode
BaseEncode
BaseForm
BaseStyle
BatesDistribution
BattleLemarieWavelet
BayesianMaximization
BayesianMaximizationObject
BayesianMinimization
BayesianMinimizationObject
BeatFrequency
Because
BeckmannDistribution
Begin
BeginPackage
Bell
BellB
BellNumber
BellY
BenfordDistribution
BeniniDistribution
BenktanderGibratDistribution
Bernoulli
BernoulliB
BernoulliDistribution
BernoulliGraphDistribution
BernoulliNumber
BernoulliProcess
BernsteinBasis
BesagL
BesselI
BesselJ
BesselJZero
BesselK
BesselY
BesselYZero
Beta
BetaBinomialDistribution
BetaDistribution
BetaNegativeBinomialDistribution
BetaPrimeDistribution
BetaRegularized
Between
BetweennessCentrality
BeveledPolyhedron
BezierCurve
BezierFunction
BezierSurface
BilateralFilter
BilateralHypergeometricPFQ
BilateralLaplaceTransform
BilateralZTransform
BinCounts
BinLists
Binarize
BinaryDeserialize
BinaryDistance
BinaryImageQ
BinaryRead
BinaryReadList
BinarySerialize
BinaryWrite
BinnedVariogramList
Binomial
BinomialDistribution
BinomialPointProcess
BinomialProcess
BinormalDistribution
BioSequence
BioSequenceBackTranslateList
BioSequenceComplement
BioSequenceInstances
BioSequenceModify
BioSequencePlot
BioSequenceQ
BioSequenceReverseComplement
BioSequenceTranscribe
BioSequenceTranslate
BiorthogonalSplineWavelet
BipartiteGraphQ
BipartiteQ
BirnbaumImportance
Bisection
BitAnd
BitClear
BitFlip
BitGet
BitLength
BitNot
BitOr
BitSet
BitShiftLeft
BitShiftRight
BitXor
BiweightLocation
BiweightMidvariance
Blend
BlockDiagonalMatrix
BlockLowerTriangularMatrix
BlockMap
BlockRandom
BlockUpperTriangularMatrix
BlomqvistBeta
BlomqvistBetaTest
Blur
Blurring
BodePlot
BohrMagneton
BohrRadius
BoltzmannConstant
BondPrice
BondYieldApprox
Boole
BooleanConsecutiveFunction
BooleanConvert
BooleanCountingFunction
BooleanFunction
BooleanGraph
BooleanMaxterms
BooleanMinimize
BooleanMinterms
BooleanQ
BooleanRegion
BooleanStrings
BooleanTable
BooleanVariables
BorderDimensions
BorelTannerDistribution
BottomHatTransform
BoundaryDiscretizeGraphics
BoundaryDiscretizeRegion
BoundaryMesh
BoundaryMeshRegion
BoundaryMeshRegionQ
BoundaryStyle
BoundedRegionQ
BoundingRegion
BoxMatrix
BoxRatios
BoxStyle
BoxWhiskerChart
Boxed
Bra
BraKet
BraggLaw
BrayCurtisDistance
BreadthFirstScan
BrightnessEqualize

BrownForsytheTest
BrownianBridgeProcess
BubbleChart
BubbleChart3D
BubbleHistogram
BubbleScale
BubbleSizes
BuckyballGraph
BulletGauge
BunchKaufmanDecomposition
ButterflyGraph
ByteArray
ByteArrayFormat
ByteArrayFormatQ
ByteArrayQ
ByteArrayToString
ByteCount

C

CAGR
CDF
CDFWavelet
CForm
CMYKColor
CNF
CSGRegion
CSGRegionQ
CSGRegionTree
CaesarCipher
Callout
CalloutMarker
CalloutStyle
CanberraDistance
CandlestickChart
CanonicalGraph
CanonicalWarpingCorrespondence
CanonicalWarpingDistance
CanonicalizePolygon
CanonicalizePolyhedron
CanonicalizeRegion
CantorMesh
CantorStaircase
Cap
CapForm
Capacitance
CapacitorEnergy
CapitalDifferentialD
Capitalize
CapsuleShape
CaputoD
Cardinality
CarlsonRC
CarlsonRD
CarlsonRE
CarlsonRF
CarlsonRG
CarlsonRJ
CarlsonRK
CarlsonRM
CarmichaelLambda
CarnotEfficiency
CartesianProduct
Casoratian
Cast
CastColumns
Catalan
CatalanNumber
CatchExceptions
CategoricalDistribution
CategoricalHistogram
CategoricalValue
Catenate
CauchyDistribution
CauchyMatrix
CauchyPointProcess
CayleyGraph
Ceiling
CellularAutomaton
CensoredDistribution
Censoring
CenterArray
CenterDot
CenteredInterval
CentralFeature
CentralMoment
CentralMomentGeneratingFunction
CentripetalForce
Cepstrogram
CepstrogramArray
CepstrumArray
ChampernowneNumber
ChanVeseBinarize
CharacterCounts
CharacterName
CharacterNormalize
CharacterRange
CharacteristicFunction
CharacteristicPolynomial
Characters
ChartBaseStyle
ChartElementFunction
ChartElements
ChartLabels
ChartLayout
ChartLegends
ChartStyle
ChebyshevT
ChebyshevU
Check
CheckAbort
ChessboardDistance
ChiDistribution
ChiSquareDistribution
ChiSquaredDistribution
ChiSquaredTest
ChineseRemainder
CholeskyDecomposition
Chop
ChromaticNumber
ChromaticPolynomial
ChromaticityPlot
ChromaticityPlot3D
Chunk
Circle
CircleArea
CircleCircumference
CircleDot
CircleMinus
CirclePlus
CirclePoints
CircleThrough
CircleTimes
CirculantGraph
CircularArcThrough
CircularOrthogonalMatrixDistribution
CircularQuaternionMatrixDistribution
CircularRealMatrixDistribution
CircularSymplecticMatrixDistribution
CircularUnitaryMatrixDistribution
CircumscribedBall
Circumsphere
ClearDistributedDefinitions
ClearSystemCache
ClebschGordan
CliffordAlgebra
Clip
ClipPlanes
ClipPlanesStyle
ClipRange
ClippingStyle
Clock
ClockGauge
Close
CloseKernels
ClosenessCentrality
Closing
ClusterDissimilarityFunction
ClusteringCoefficient
ClusteringComponents
ClusteringMeasurements
ClusteringTree
Coefficient
CoefficientArrays
CoefficientList
CoefficientOfVariation
CoefficientRules
CohensD
CoifletWavelet
CollatzLength
CollatzSequence
Collect
CollinearPoints
Colon
ColorBalance
ColorCombine
ColorConvert
ColorCoverage
ColorDataFunction
ColorDetect
ColorDistance
ColorFunction
ColorFunctionBinning
ColorFunctionScaling
ColorNegate
ColorQ
ColorQuantize
ColorReplace
ColorRules
ColorSeparate
ColorSpace
ColorToneMapping
Colorize
ColorsNear
Column
ColumnKeyExistsQ
ColumnKeys
ColumnSpace
ColumnTypes
ColumnwiseCombine
ColumnwiseThread
ColumnwiseValue

Comap
ComapApply
CombinatorB
CombinatorC
CombinatorI
CombinatorK
CombinatorS
CombinatorW
CombinatorY
Commonest
CommonestFilter
CommunityBoundaryStyle
CommunityGraphPlot
CommunityLabels
CommunityRegionStyle
Commutator
CompanionMatrix
Compile
Compile`ConstructExpression
Compile`CreateTypeInstance
Compile`FromRawPointer
Compile`InertConstruct
Compile`InertEvaluate
Compile`InertOperation
Compile`KernelEvaluate
Compile`ToRawPointer
CompiledCodeFunction
CompiledExpressionDeclaration
CompiledFunction
CompilerEnvironmentAppendTo
CompilerEnvironmentObject
Complement
CompleteGraph
CompleteGraphQ
CompleteIntegral
CompleteKaryTree
CompleteTheSquare
ComplexArrayPlot
ComplexContourPlot
ComplexExpand
ComplexListPlot
ComplexPlot
ComplexPlot3D
ComplexRegionPlot
ComplexStreamPlot
ComplexVectorPlot
ComplexityFunction
ComponentExpand
ComponentMeasurements
ComposeList
ComposeSeries
CompositeQ
Compositions
CompoundInterest
CompoundPoissonDistribution
CompoundPoissonProcess
CompoundRenewalProcess
Compress
ComptonWavelength
ConcaveHullMesh
ConditionNumber
ConditionalExpression
Conditioned
Cone
ConeSurfaceArea
ConeVolume
ConfidenceInterval
ConfidenceLevel
ConfidenceRange
ConfidenceTransform
Confirm
ConfirmAssert
ConfirmBy
ConfirmMatch
ConfirmQuiet
ConformImages
Congruent
ConicGradientFilling
ConicHullRegion
ConicOptimization
Conjugate
ConjugateTranspose
Conjunction
ConnectedComponents
ConnectedGraphComponents
ConnectedGraphQ
ConnectedMeshComponents
ConnectedQ
ConoverTest
ConservativeConvectionPDETerm
ConstantArray
ConstantImage
ConstantRegionQ
Construct
ConstructColumns
ContainsAll
ContainsAny
ContainsExactly
ContainsNone
ContainsOnly
ContentObject
ContentSize
Contexts
ContinuedFraction
ContinuedFractionConvergents
ContinuedFractionK
ContinuousCompounding
ContinuousMarkovProcess
ContinuousWaveletData
ContinuousWaveletTransform
ContourDetect
ContourIntegrate
ContourLabels
ContourPlot
ContourPlot3D
ContourShading
ContourStyle
Contours
ContraharmonicMean
ConvectionPDETerm
Convergents
ConvexHullMesh
ConvexHullRegion
ConvexOptimization
ConvexPolygonQ
ConvexPolyhedronQ
ConvexRegionQ
Convolution
Convolve
ConwayGroupCo1
ConwayGroupCo2
ConwayGroupCo3
CoordinateBoundingBoxArray
CoordinateBounds
CoordinateBoundsArray
CoordinateTransform
CoplanarPoints
CoprimeQ
Coproduct
CopulaDistribution
CopyDirectory
CopyFile
CoreNilpotentDecomposition
CornerFilter
Correlation
CorrelationDistance
CorrelationFunction
CorrelationTest
Cos
CosIntegral
Cosh
CoshIntegral
CosineDistance
Cot
Coth
CoulombConstant
CoulombF
CoulombForce
CoulombG
CoulombH1
CoulombH2
CountDistinct
CountDistinctBy
CountRoots
Counts
CountsBy
Covariance
CovarianceEstimatorFunction
CovarianceFunction
CoxIngersollRossProcess
CoxModel
CoxModelFit
CoxianDistribution
CramerVonMisesTest
CreateArchive
CreateCompilerEnvironment
CreateDataStructure
CreateDirectory
CreateFile
CreateForeignCallback
CreateManagedObject
CreateSearchIndex
CreateTypeInstance
CreateUUID
CriterionFunction
CriticalPoints
CriticalSection
Cross
Cross3D
CrossMatrix
CrossProduct
CrossingCount
CrossingDetect
CrossingPolygon
Csc
Csch
Cube
CubeRoot
CubeSurfaceArea
CubeVolume
Cubics

Cuboid
Cumulant
CumulantGeneratingFunction
Cup
CupCap
Curl
CurryApplied
CurvatureFlowFilter
CurveClosed
CycleGraph
CycleIndexPolynomial
Cycles
Cyclic
CyclicGroup
Cyclotomic
Cylinder
CylinderSurfaceArea
CylinderVolume
CylindricalDecomposition
CylindricalDecompositionFunction

D

DCT
DEigensystem
DEigenvalues
DFT
DFixedPoints
DGaussianWavelet
DMSList
DMSString
DNF
DST
DSolveChangeVariables
DSolveValue
DStabilityConditions
DagumDistribution
DamerauLevenshteinDistance
Darker
Dashing
DataConnectionObject
DataRange
DataReversed
DataStructure
DataStructureQ
DatabaseConnect
DatabaseDisconnect
DatabaseReference
Dataset
DateBounds
DateDifference
DateHistogram
DateInterval
DateList
DateListLogPlot
DateListPlot
DateListStepPlot
DateObject
DateObjectQ
DateOverlapsQ
DatePattern
DateScale
DateString
DateTicksFormat
DateTimeString
DateWithinQ
Dated
DaubechiesWavelet
DavisDistribution
DawsonF
DayOfWeek
DaysInMonth
DeBroglieWavelength
DeBruijnGraph
DeBruijnSequence
Decapitalize
DecimalForm
DecisionTreeTrain
DeclarePackage
Decompose
Decrypt
DecryptFile
DedekindEta
Default
DefaultAxesStyle
DefaultBaseStyle
DefaultFaceGridsStyle
DefaultFrameStyle
DefaultFrameTicksStyle
DefaultGridLinesStyle
DefaultLabelStyle
DefaultTicksStyle
DefineInputStreamMethod
DefineOutputStreamMethod
DefiniteIntegral
Degree
DegreeCentrality
DegreeGraphDistribution
Del
DelaunayMesh
Delete
DeleteAdjacentDuplicates
DeleteBorderComponents
DeleteColumns
DeleteDirectory
DeleteDuplicates
DeleteDuplicatesBy
DeleteElements
DeleteFile
DeleteMissing
DeleteObject
DeleteSearchIndex
DeleteSmallComponents
Dendrogram
Denominator
DensityHistogram
DensityPlot
DensityPlot3D
Depth
DepthFirstScan
Derangements
Derivative
DerivativeFilter
DerivativePDETerm
DerivedKey
DesignMatrix
DiGamma
Diagonal
DiagonalMatrix
DiagonalMatrixQ
DiagonalizableMatrixQ
Diamond
DiamondMatrix
DiceDissimilarity
Diff3
DiffApply
DifferenceDelta
DifferenceQuotient
DifferenceRootReduce
Differences
DifferentialD
DifferentialRootReduce
DifferentiatorFilter
DiffusionPDETerm
DiggleGatesPointProcess
DiggleGrattonPointProcess
DigitCount
DigitQ
DigitSum
DigitalSignature
DihedralAngle
DihedralGroup
Dijkstra
Dilation
DimensionalMeshComponents
Dimensions
DiophantineSolve
DiracComb
DiracDelta
DirectedEdge
DirectedGraph
DirectedGraphQ
DirectedInfinity
DirectionalDerivative
DirectionalLight
Directive
Directory
DirectoryName
DirectoryQ
DirectoryStack
DirichletBeta
DirichletCharacter
DirichletConvolve
DirichletDistribution
DirichletEta
DirichletL
DirichletLambda
DirichletTransform
DisableFormatting
Discard
DiscreteAsymptotic
DiscreteChirpZTransform
DiscreteConvolve
DiscreteCosineTransform
DiscreteDelta
DiscreteFourierTransform
DiscreteHadamardTransform
DiscreteHilbertTransform
DiscreteIndicator
DiscreteLimit
DiscreteLyapunovSolve
DiscreteMarkovProcess
DiscreteMaxLimit
DiscreteMinLimit
DiscretePlot
DiscretePlot3D
DiscreteRatio
DiscreteRiccatiSolve
DiscreteShift
DiscreteSineTransform
DiscreteUniformDistribution
DiscreteWaveletData
DiscreteWaveletPacketTransform

DiscreteWaveletTransform
DiscretizeGraphics
DiscretizeRegion
Discriminant
DisjointQ
Disjunction
Disk
DiskMatrix
DiskSegment
Dispatch
DispersionEstimatorFunction
Distance
Distance3D
DistanceMatrix
DistanceTransform
Distribute
DistributeDefinitions
Distributed
DistributionChart
DistributionFitTest
DistributionParameterAssumptions
DistributionParameterQ
Dithering
Div
Divergence
DivideSides
Divisible
DivisorCount
DivisorSigma
DivisorSum
Divisors
Dodecahedron
DominantColors
DominatorTreeGraph
DominatorVertexList
DopplerShift
DotEqual
DoubleBracketingBar
DoubleDownArrow
DoubleFactorial
DoubleLeftArrow
DoubleLeftRightArrow
DoubleLeftTee
DoubleLongLeftArrow
DoubleLongLeftRightArrow
DoubleLongRightArrow
DoubleRightArrow
DoubleRightTee
DoubleUpArrow
DoubleUpDownArrow
DoubleVerticalBar
DoublingTime
DownArrow
DownArrowBar
DownArrowUpArrow
DownLeftRightVector
DownLeftTeeVector
DownLeftVector
DownLeftVectorBar
DownRightTeeVector
DownRightVector
DownRightVectorBar
DownTee
DownTeeArrow
DownValuesFunction
DrazinInverse
DropShadowing
Dt
DualPlanarGraph
DualPolyhedron
DumpSave
DuplicateFreeQ

E

EarthMass
EarthRadius
EccentricityCentrality
Echo
EchoEvaluation
EchoFunction
EchoLabel
EchoTiming
EdgeAdd
EdgeBetweennessCentrality
EdgeCapacity
EdgeChromaticNumber
EdgeConnectivity
EdgeContract
EdgeCount
EdgeCoverQ
EdgeCycleMatrix
EdgeDelete
EdgeDetect
EdgeForm
EdgeIndex
EdgeList
EdgeQ
EdgeRules
EdgeTaggedGraph
EdgeTaggedGraphQ
EdgeTags
EdgeTransitiveGraphQ
EdgeValueRange
EdgeValueSizes
EdgeWeight
EdgeWeightedGraphQ
EditDistance
EffectSize
EffectiveRate
Eigensystem
EigenvalueDecomposition
Eigenvalues
EigenvectorCentrality
Eigenvectors
ElectricCurrentDensityValue
ElectricCurrentPDEComponent
ElectricField
ElectricFluxDensityValue
ElectricPotential
ElectricPotentialCondition
ElectricSymmetryValue
ElectronMass
ElectrostaticPDEComponent
Element
ElementQ
ElementaryCharge
Elementwise
Eliminate
Ellipsoid
EllipsoidVolume
EllipticE
EllipticEIncomplete
EllipticExp
EllipticExpPrime
EllipticF
EllipticK
EllipticLog
EllipticNomeQ
EllipticPi
EllipticTheta
EllipticThetaPrime
EmpiricalDistribution
EmptyGraphQ
EmptyRegion
EmptySpaceF
Enclose
Encrypt
EncryptFile
EncryptedObject
End
EndPackage
EngineeringForm
Entropy
EntropyChange
EntropyFilter
Environment
Epilog
EpilogFunction
EqualTilde
Equilibrium
Equivalent
EquivalentStrain
Erf
Erfc
Erfi
ErlangB
ErlangC
ErlangDistribution
Erosion
EscapeRadius
EscapeVelocity
EstimatedBackground
EstimatedDistribution
EstimatedPointNormals
EstimatedPointProcess
EstimatedProcess
EstimatedVariogramModel
EuclideanDistance
EulerAngles
EulerCharacteristic
EulerE
EulerMatrix
EulerMethod
EulerNumber
EulerPhi
EulerianGraphQ
EventLabels
EventSeries
EventSeriesAccumulate
EventSeriesQ
ExactNumberQ
ExactODEQ
Exception
ExceptionQ
ExceptionTypeRegisteredQ
ExceptionTypes
ExcludedForms
ExcludedLines
Exclusions
ExclusionsStyle
Exists
Exit

Exp
ExpGammaDistribution
ExpIntegralE
ExpIntegralEi
ExpToTrig
Expand
ExpandAll
ExpandDenominator
ExpandFileName
ExpandNumerator
Expectation
Exponent
ExponentialDistribution
ExponentialFamily
ExponentialGeneratingFunction
ExponentialModel
ExponentialMovingAverage
ExponentialPowerDistribution
Export
ExportByteArray
ExportString
ExpressionGraph
ExpressionTree
ExtendedGCD
ExtendedKey
ExtentElementFunction
ExtentMarkers
ExtentSize
Extract
ExtractArchive
ExtremeValueDistribution

F

FARIMAProcess
FDistribution
FFT
FRatioDistribution
FaceForm
FaceGrids
FaceGridsStyle
Factor
FactorInteger
FactorList
FactorSquareFree
FactorSquareFreeList
FactorTerms
FactorTermsList
Factorial
Factorial2
FactorialMoment
FactorialMomentGeneratingFunction
FactorialPower
Failsafe
Failure
FailureAction
FailureDistribution
FailureQ
FaradayConstant
FaradayLaw
FareySequence
FeatureDistance
Fibonacci
FibonacciNumber
FibonacciSequence
Fibonorial
FileBaseName
FileByteCount
FileConvert
FileDate
FileExistsQ
FileExtension
FileFormat
FileFormatProperties
FileFormatQ
FileHash
FileNameDepth
FileNameDrop
FileNameJoin
FileNameSplit
FileNameTake
FileNameToFormatList
FileNames
FilePrint
FileSize
FileSystemMap
FileSystemScan
FileSystemTree
FileTemplate
FileTemplateApply
FileType
FilledCurve
FilledPolarCurve
FilledTorus
Filling
FillingStyle
FillingTransform
FilterRules
Find
FindArgMax
FindArgMin
FindClique
FindClusters
FindCookies
FindCurvePath
FindCycle
FindDistribution
FindDistributionParameters
FindDivisions
FindEdgeColoring
FindEdgeCover
FindEdgeCut
FindEdgeIndependentPaths
FindEquationalProof
FindEulerianCycle
FindFile
FindFit
FindFundamentalCycles
FindGeneratingFunction
FindGeometricConjectures
FindGeometricTransform
FindGraphCommunities
FindGraphIsomorphism
FindGraphPartition
FindHamiltonianCycle
FindHamiltonianPath
FindHiddenMarkovStates
FindIndependentEdgeSet
FindIndependentVertexSet
FindInstance
FindIntegerNullVector
FindIsomorphicSubgraph
FindKClan
FindKClique
FindKClub
FindKPlex
FindLinearRecurrence
FindList
FindMatchingColor
FindMatrixGameStrategies
FindMaxValue
FindMaximum
FindMaximumCut
FindMaximumFlow
FindMeshDefects
FindMinValue
FindMinimum
FindMinimumCostFlow
FindMinimumCut
FindPath
FindPeaks
FindPermutation
FindPlanarColoring
FindPointProcessParameters
FindPostmanTour
FindProcessParameters
FindRegionTransform
FindRepeat
FindRoot
FindSequenceFunction
FindShortestCurve
FindShortestPath
FindShortestTour
FindSpanningTree
FindSubgraphIsomorphism
FindThreshold
FindTransientRepeat
FindTreeGameStrategies
FindVertexColoring
FindVertexCover
FindVertexCut
FindVertexIndependentPaths
FineStructureConstant
FiniteAbelianGroupCount
FiniteField
FiniteFieldElement
FiniteFieldElementNorm
FiniteFieldElementPrimitiveQ
FiniteFieldElementTrace
FiniteFieldEmbedding
FiniteFieldIndex
FiniteGroupCount
FirstCase
FirstPassageTimeDistribution
FirstPosition
FischerGroupFi22
FischerGroupFi23
FischerGroupFi24Prime
FisherHypergeometricDistribution
FisherRatioTest
FisherZDistribution
Fit
FitDegree
FittedModel
FlatShading
FlattenAt
Floor
FlowPolynomial
FluidFlowPDEComponent
FluidViscosity
FluidViscousStress
FoldPair
FoldPairList

FoldWhile
FoldWhileList
ForAll
ForAllType
Force
ForeignCallback
ForeignFunction
ForeignFunctionLoad
ForeignPointerLookup
FortranForm
Fourier
FourierCoefficient
FourierCosCoefficient
FourierCosSeries
FourierCosTransform
FourierDCT
FourierDCTFilter
FourierDCTMatrix
FourierDFT
FourierDST
FourierDSTMatrix
FourierMatrix
FourierSequenceTransform
FourierSeries
FourierSinCoefficient
FourierSinSeries
FourierSinTransform
FourierTransform
FourierTrigSeries
FoxH
FoxHReduce
FractionalBrownianMotionProcess
FractionalD
FractionalGaussianNoiseProcess
FractionalPart
Frame
FrameLabel
FrameMargins
FrameStyle
FrameTicks
FrameTicksStyle
Framed
FrechetDistribution
FreeFallDistance
FrenetSerretSystem
FresnelC
FresnelF
FresnelG
FresnelS
FrobeniusAutomorphism
FrobeniusDecomposition
FrobeniusNorm
FrobeniusNumber
FrobeniusReduce
FrobeniusSolve
FromCharacterCode
FromCoefficientRules
FromContinuedFraction
FromDMS
FromDigits
FromFiniteField
FromFiniteFieldIndex
FromJulianDate
FromLetterNumber
FromLunationNumber
FromPackedArray
FromPolarCoordinates
FromRawPointer
FromRomanNumeral
FromSphericalCoordinates
FromTabular
FrustumVolume
FullGraphics
FullRegion
FullSimplify
FunctionAnalytic
FunctionBijective
FunctionCompile
FunctionContinuous
FunctionConvexity
FunctionDeclaration
FunctionDiscontinuities
FunctionDomain
FunctionExpand
FunctionInjective
FunctionInterpolation
FunctionMeromorphic
FunctionMonotonicity
FunctionPeriod
FunctionPoles
FunctionRange
FunctionSign
FunctionSingularities
FunctionSpace
FunctionSurjective
FutureValue

G

GARCHProcess
GCD
GaborFilter
GaborMatrix
GaborWavelet
Gamma
GammaDistribution
GammaRegularized
GasConstant
Gather
GatherBy
GaugeFaceElementFunction
GaugeFaceStyle
GaugeFrameElementFunction
GaugeFrameSize
GaugeFrameStyle
GaugeLabels
GaugeMarkers
GaugeStyle
GaussQuadrature
GaussianFilter
GaussianMatrix
GaussianOrthogonalMatrixDistribution
GaussianSymplecticMatrixDistribution
GaussianUnitaryMatrixDistribution
GegenbauerC
GeneralizedLinearModelFit
GeneralizedPolyLog
GeneralizedPower
GenerateAsymmetricKeyPair
GenerateConditions
GenerateDerivedKey
GenerateDigitalSignature
GenerateFileSignature
GenerateSymmetricKey
GeneratingFunction
GenericCylindricalDecomposition
GeoDestination
GeoDirection
GeoDisk
GeoDisplacement
GeoDistance
GeoPath
GeodesicClosing
GeodesicDilation
GeodesicErosion
GeodesicOpening
GeodesicPolyhedron
GeometricAssertion
GeometricBrownianMotionProcess
GeometricDistribution
GeometricMean
GeometricMeanFilter
GeometricOptimization
GeometricScene
GeometricSequence
GeometricSolveValues
GeometricStep
GeometricStylingRules
GeometricTest
GeometricTransformation
Get
GetEnvironment
GibbsPointProcess
GlobalClusteringCoefficient
Glow
GompertzMakehamDistribution
GoochShading
GoodmanKruskalGamma
GoodmanKruskalGammaTest
Goto
GouraudShading
Grad
Gradient
GradientFilter
GradientFittedMesh
GradientOrientationFilter
Graph
Graph3D
GraphAssortativity
GraphAutomorphismGroup
GraphCenter
GraphComplement
GraphDensity
GraphDiameter
GraphDifference
GraphDisjointUnion
GraphDistance
GraphDistanceMatrix
GraphEmbedding
GraphHub
GraphIntersection
GraphJoin
GraphLayerStyle
GraphLayers
GraphLinkEfficiency
GraphPeriphery
GraphPlot
GraphPlot3D
GraphPower
GraphProduct
GraphPropertyDistribution

GraphQ
GraphRadius
GraphReciprocity
GraphSum
GraphTree
GraphTriangleCount
GraphUnion
GraphValuePlot
Graphics
Graphics3D
GraphicsColumn
GraphicsComplex
GraphicsGrid
GraphicsGroup
GraphicsRow
GrassmannAlgebra
GravitationalConstant
GrayLevel
GreaterEqualLess
GreaterFullEqual
GreaterGreater
GreaterLess
GreaterSlantEqual
GreaterTilde
GreenFunction
Grid
GridGraph
GridLines
GridLinesStyle
GroebnerBasis
GroupBy
GroupCentralizer
GroupElementFromWord
GroupElementPosition
GroupElementQ
GroupElementToWord
GroupElements
GroupGenerators
GroupMultiplicationTable
GroupOrbits
GroupOrder
GroupSetwiseStabilizer
GroupStabilizer
GroupStabilizerChain
Groupings
GrowCutComponents
GrowingAnnuityPV
GrowingPerpetuityPV
Gudermannian
GuidedFilter
GumbelDistribution

H

HITSCentrality
HTTPRedirect
HTTPRequest
HTTPResponse
HaarWavelet
HadamardMatrix
HalfLine
HalfNormalDistribution
HalfPlane
HalfSpace
HalftoneShading
Haloing
HamiltonianGraphQ
HammingDistance
HandlerFunctions
HandlerFunctionsKeys
HankelH1
HankelH2
HankelMatrix
HankelTransform
HaradaNortonGroupHN
HararyGraph
HardcorePointProcess
HarmonicMean
HarmonicMeanFilter
HarmonicNumber
HarmonicPolyLog
Hash
HatchFilling
HatchShading
HazardFunction
HeatCapacity
HeatFluxValue
HeatInsulationValue
HeatOutflowValue
HeatRadiationValue
HeatSymmetryValue
HeatTemperatureCondition
HeatTransferPDEComponent
HeatTransferValue
Heatmap
HeavisideLambda
HeavisidePi
HeavisideTheta
HeldGroupHe
HelmholtzPDEComponent
HermiteDecomposition
HermiteH
HermiteReduce
Hermitian
HermitianMatrix
HermitianMatrixQ
HermitianQ
HessenbergDecomposition
HessianMatrix
HeunB
HeunBPrime
HeunC
HeunCPrime
HeunD
HeunDPrime
HeunG
HeunGPrime
HeunT
HeunTPrime
HexDecode
HexEncode
Hexahedron
HiddenMarkovProcess
HighlightGraph
HighlightImage
HighlightMesh
HighlightRegion
Highlighted
HighpassFilter
HigmanSimsGroupHS
HilbertCurve
HilbertFilter
HilbertMatrix
HilbertTransform
Histogram
Histogram3D
HistogramDistribution
HistogramList
HistogramPointDensity
HistogramTransform
HistogramTransformInterpolation
HitMissTransform
HodgeDual
HoeffdingD
HoeffdingDTest
HookLengthFormula
HorizontalGauge
HornerForm
HostLookup
HotellingTSquareDistribution
HoytDistribution
HubbleConstant
Hue
HumpDownHump
HumpEqual
HurwitzLerchPhi
HurwitzZeta
HyperHarmonicNumber
HyperbolicDistribution
HypercubeGraph
HyperexponentialDistribution
Hyperfactorial
Hypergeometric0F1
Hypergeometric0F1Regularized
Hypergeometric1F1
Hypergeometric1F1Regularized
Hypergeometric2F1
Hypergeometric2F1Regularized
HypergeometricDistribution
HypergeometricPFQ
HypergeometricPFQRegularized
HypergeometricU
Hyperplane
HypoexponentialDistribution
Hypotenuse

I

IPAddress
IQR
IRR
Icosahedron
IdentityMatrix
IfCompiled
Iff
Im
Image
Image3D
Image3DProjection
Image3DSlices
ImageAccumulate
ImageAdd
ImageAdjust
ImageAlign
ImageApply
ImageApplyIndexed
ImageAspectRatio
ImageAssemble
ImageChannels
ImageClip
ImageCollage
ImageColorSpace
ImageCompose
ImageConvolve
ImageCooccurrence

ImageCorrelate
ImageCorrespondingPoints
ImageCrop
ImageData
ImageDeconvolve
ImageDemosaic
ImageDifference
ImageDimensions
ImageDisplacements
ImageDistance
ImageEffect
ImageExposureCombine
ImageFeatureTrack
ImageFileApply
ImageFileFilter
ImageFileScan
ImageFilter
ImageFocusCombine
ImageForestingComponents
ImageForwardTransformation
ImageHistogram
ImageKeypoints
ImageLabels
ImageLegends
ImageLevels
ImageLines
ImageMargins
ImageMarker
ImageMeasurements
ImageMesh
ImageMultiply
ImagePad
ImagePadding
ImagePartition
ImagePeriodogram
ImagePerspectiveTransformation
ImagePyramid
ImagePyramidApply
ImageQ
ImageRecolor
ImageReflect
ImageResize
ImageResolution
ImageRotate
ImageScaled
ImageScan
ImageSegmentationComponents
ImageSegmentationFilter
ImageSize
ImageSizeAction
ImageSizeMultipliers
ImageStitch
ImageSubtract
ImageTake
ImageTransformation
ImageTrim
ImageType
ImageValue
ImageValuePositions
ImageVectorscopePlot
ImageWaveformPlot
ImaginaryPart
ImplicitD
ImplicitPlot
ImplicitRegion
Implies
Import
ImportByteArray
ImportString
Impulse
InString
IncidenceGraph
IncidenceList
IncidenceMatrix
IncludeDirectories
IncludeGroupAggregates
IncludeSingularSolutions
IncrementalFunction
IncrementalObject
IncrementalReceive
IncrementalYield
IndefiniteMatrixQ
IndependenceTest
IndependentEdgeSetQ
IndependentVertexSetQ
IndexEdgeTaggedGraph
IndexGraph
Indexed
InductanceEnergy
InertEvaluate
InertExpression
InexactNumberQ
InfiniteLine
InfiniteLineThrough
InfinitePlane
InflectionPoints
InformationData
InhomogeneousPoissonPointProcess
InhomogeneousPoissonProcess
InnerPolygon
InnerPolyhedron
Inpaint
Input
InputString
InscribedBall
Insert
InsertColumns
InsertLinebreaks
Inset
Insphere
IntegerDigits
IntegerExponent
IntegerLength
IntegerPart
IntegerPartitions
IntegerReverse
IntegerString
Integrate
IntegrateChangeVariables
IntegratingFactor
Interactive3D
Interleave
Interleaving
IntermediateTest
InterpolatingPolynomial
Interpolation
InterquartileRange
Interrupt
IntersectingQ
Intersection
Interval
IntervalIntersection
IntervalMarkers
IntervalMarkersStyle
IntervalMemberQ
IntervalUnion
InverseBetaRegularized
InverseBilateralLaplaceTransform
InverseBilateralZTransform
InverseCDF
InverseChiSquareDistribution
InverseContinuousWaveletTransform
InverseDistanceTransform
InverseEllipticNomeQ
InverseErf
InverseErfc
InverseFFT
InverseFourier
InverseFourierCosTransform
InverseFourierSequenceTransform
InverseFourierSinTransform
InverseFourierTransform
InverseFunction
InverseGammaDistribution
InverseGammaRegularized
InverseGaussianDistribution
InverseGudermannian
InverseHankelTransform
InverseHilbertTransform
InverseImagePyramid
InverseJacobiCD
InverseJacobiCN
InverseJacobiCS
InverseJacobiDC
InverseJacobiDN
InverseJacobiDS
InverseJacobiNC
InverseJacobiND
InverseJacobiNS
InverseJacobiSC
InverseJacobiSD
InverseJacobiSN
InverseLaplaceTransform
InverseMellinTransform
InversePermutation
InverseRadonTransform
InverseSeries
InverseShortTimeFourier
InverseSpectrogram
InverseSurvivalFunction
InverseTransformedRegion
InverseWaveletTransform
InverseWeierstrassP
InverseWishartMatrixDistribution
InverseZTransform
IrreduciblePolynomialQ
IsPrime
IsolatingInterval
IsomorphicGraphQ
IsomorphicSubgraphQ
Item
ItoProcess

J

JaccardDissimilarity
JacobiAmplitude
JacobiCD
JacobiCN
JacobiCS
JacobiDC
JacobiDN
JacobiDS

JacobiEpsilon
JacobiNC
JacobiND
JacobiNS
JacobiP
JacobiSC
JacobiSD
JacobiSN
JacobiSymbol
JacobiZN
JacobiZeta
JacobianMatrix
JankoGroupJ1
JankoGroupJ2
JankoGroupJ3
JankoGroupJ4
JarqueBeraALMTest
JohnsonDistribution
JoinAcross
JoinForm
Joined
JoinedCurve
JordanDecomposition
JordanForm
JordanMatrix
JordanReduce
JordanTotient
JuliaSetBoettcher
JuliaSetIterationCount
JuliaSetPlot
JuliaSetPoints
JulianDate

K

KCoreComponents
KDistribution
KEdgeConnectedComponents
KEdgeConnectedGraphQ
KMeans
KVertexConnectedComponents
KVertexConnectedGraphQ
KagiChart
KalmanFilter
KarhunenLoeveDecomposition
KaryTree
KatzCentrality
KelvinBei
KelvinBer
KelvinKei
KelvinKer
KendallCorrelation
KendallTau
KendallTauTest
KernelEvaluate
KernelFunction
KernelMixtureDistribution
KernelModelFit
Kernels
Ket
KeyComplement
KeyDrop
KeyDropFrom
KeyExistsQ
KeyFreeQ
KeyIntersection
KeyMap
KeyMemberQ
KeySelect
KeySort
KeySortBy
KeyTake
KeyUnion
KeyValueMap
Keys
KillProcess
KineticEnergy
KirchhoffGraph
KirchhoffMatrix
KleinInvariantJ
KnapsackSolve
KnightTourGraph
KochCurve
KolmogorovSmirnovTest
KroneckerDelta
KroneckerProduct
KroneckerSymbol
KruskalWallis
KuiperTest
KumaraswamyDistribution
Kurtosis
KuwaharaFilter

L

LABColor
LCFrequency
LCHColor
LCM
LDLDecomposition
LUDecomposition
LUVColor
LabelStyle
LabelVisibility
Labeled
LabelingFunction
LabelingSize
LabelingTarget
LaguerreL
LambdaComponents
LambertW
LameC
LameCPrime
LameEigenvalueA
LameEigenvalueB
LameS
LameSPrime
LandauDistribution
LaplaceDistribution
LaplaceODESolve
LaplaceTransform
Laplacian
LaplacianFilter
LaplacianGaussianFilter
LaplacianPDETerm
Latitude
LatitudeLongitude
LatticeReduce
LaunchKernels
LaurentSeries
LayerSizeFunction
LayeredGraph
LayeredGraph3D
LayeredGraphPlot
LayeredGraphPlot3D
LeaderSize
LeafCount
LeapYearQ
LeastSquares
LeftArrow
LeftArrowBar
LeftArrowRightArrow
LeftDownTeeVector
LeftDownVector
LeftDownVectorBar
LeftJoinAcross
LeftRightArrow
LeftRightVector
LeftTee
LeftTeeArrow
LeftTeeVector
LeftTriangle
LeftTriangleBar
LeftTriangleEqual
LeftUpDownVector
LeftUpTeeVector
LeftUpVector
LeftUpVectorBar
LeftVector
LeftVectorBar
LegendAppearance
LegendFunction
LegendLabel
LegendLayout
LegendMargins
LegendMarkerSize
LegendMarkers
Legended
LegendreP
LegendreQ
LegendreSymbol
LengthContraction
LengthWhile
LerchPhi
LessEqualGreater
LessFullEqual
LessGreater
LessLess
LessSlantEqual
LessTilde
LetterCounts
LetterNumber
LetterQ
Level
LeveneTest
LeviCivitaTensor
LevyDistribution
LexicographicOrder
LexicographicSort
LiftingWaveletTransform
LightYear
Lighter
Lighting
LightingAngle
Likelihood
LindleyDistribution
Line
LineGraph
LineIntegralConvolutionPlot
LineIntegralConvolutionScale
LineIntegrate
LineLegend
LineSphereIntersection
LinearFractionalOptimization
LinearFractionalTransform

LinearGradientFilling
LinearGradientImage
LinearModel
LinearModelFit
LinearOffsetFunction
LinearOptimization
LinearProgramming
LinearRecurrence
LinearRegression
LinearSolve
LinearSolveFunction
LinkFunction
LinkRankCentrality
LiouvilleLambda
ListContourPlot
ListContourPlot3D
ListConvolve
ListCorrelate
ListCurvePathPlot
ListDeconvolve
ListDensityPlot
ListDensityPlot3D
ListFitPlot
ListFitPlot3D
ListFourierSequenceTransform
ListInterpolation
ListLineIntegralConvolutionPlot
ListLinePlot
ListLinePlot3D
ListLogLinearPlot
ListLogLogPlot
ListLogPlot
ListPlot
ListPlot3D
ListPointPlot3D
ListPolarPlot
ListSliceContourPlot3D
ListSliceDensityPlot3D
ListSliceVectorPlot3D
ListStepPlot
ListStreamDensityPlot
ListStreamPlot
ListStreamPlot3D
ListSurfacePlot3D
ListVectorDensityPlot
ListVectorDisplacementPlot
ListVectorDisplacementPlot3D
ListVectorPlot
ListVectorPlot3D
ListZTransform
LiteralType
LoanPayment
LocalAdaptiveBinarize
LocalCache
LocalClusteringCoefficient
LocalEvaluate
LocalModelFit
LocalSubmit
LocalSymbol
LocationEquivalenceTest
LocationTest
Log
Log10
Log2
LogBarnesG
LogGamma
LogGammaDistribution
LogIntegral
LogLikelihood
LogLinearPlot
LogLogPlot
LogLogisticDistribution
LogModel
LogMultinormalDistribution
LogNormalDistribution
LogPlot
LogRankTest
LogSeriesDistribution
LogicalExpand
LogisticDistribution
LogisticRegressionTrain
LogisticSigmoid
LogitModelFit
LommelS1
LommelS2
LommelT1
LommelT2
LongLeftArrow
LongLeftRightArrow
LongRightArrow
LongestCommonSequence
LongestCommonSequencePositions
LongestCommonSubsequence
LongestCommonSubsequencePositions
LongestOrderedSequence
Longitude
Lookup
LoopFreeGraphQ
LowerCaseQ
LowerLeftArrow
LowerRightArrow
LowerTriangularMatrix
LowerTriangularMatrixQ
LowerTriangularize
LowpassFilter
Lucas
LucasL
LuccioSamiComponents
LunationNumber
LyapunovDerivative
LyapunovSolve
LyonsGroupLy

M

MAD
MAProcess
MD5
MIMETypeToFormatList
MaClaurinSeries
MachineNumberQ
MagneticFieldIntensity
MagneticFieldWire
MagneticFluxDensity
MagneticFluxDensityValue
MagneticForce
MagneticPDEComponent
MagneticPotentialCondition
MagneticSymmetryValue
MagnetostaticPDEComponent
Magnification
Majority
ManagedObject
MandelbrotSetBoettcher
MandelbrotSetDistance
MandelbrotSetIterationCount
MandelbrotSetMemberQ
MandelbrotSetPlot
MangoldtLambda
ManhattanDistance
MannWhitneyTest
MannWhitneyU
MantissaExponent
MapApply
MapAt
MarchenkoPasturDistribution
MarcumQ
MardiaCombinedTest
MardiaKurtosisTest
MardiaSkewnessTest
MarginalDistribution
MarkovProcessProperties
Masking
MassConcentrationCondition
MassEnergyEquivalence
MassFluxValue
MassImpermeableBoundaryValue
MassOutflowValue
MassSymmetryValue
MassTransferValue
MassTransportPDEComponent
MatchingDissimilarity
MaterialShading
MaternPointProcess
MathMLForm
MathieuC
MathieuCPrime
MathieuCharacteristicA
MathieuCharacteristicB
MathieuCharacteristicExponent
MathieuGroupM11
MathieuGroupM12
MathieuGroupM22
MathieuGroupM23
MathieuGroupM24
MathieuS
MathieuSPrime
Matrices
MatrixCos
MatrixExp
MatrixForm
MatrixFunction
MatrixGame
MatrixGamePayoff
MatrixGamePlot
MatrixLog
MatrixMinimalPolynomial
MatrixNormalDistribution
MatrixNullity
MatrixPlot
MatrixPolynomialValue
MatrixPower
MatrixPropertyDistribution
MatrixQ
MatrixRank
MatrixSin
MatrixSqrt
MatrixSymbol
MatrixTDistribution
MatrixTan
Max
MaxCellMeasure
MaxColorDistance

MaxDate
MaxDetect
MaxExtraConditions
MaxFilter
MaxLimit
MaxMemoryUsed
MaxOverlapFraction
MaxPlotPoints
MaxRecursion
MaxRoots
MaxStableDistribution
MaxValue
MaximalBy
MaximalMatching
Maximize
MaxwellDistribution
McLaughlinGroupMcL
Mean
MeanAround
MeanClusteringCoefficient
MeanDegreeConnectivity
MeanDeviation
MeanFilter
MeanGraphDistance
MeanNeighborDegree
MeanPointDensity
MeanShift
MeanShiftFilter
MechanicalPower
Median
MedianDeviation
MedianFilter
MeijerG
MeijerGReduce
MeixnerDistribution
MellinConvolve
MellinTransform
MemoryAvailable
MemoryConstrained
MemoryConstraint
MemoryInUse
MengerMesh
Merge
MergingFunction
MersennePrimeExponent
MersennePrimeExponentQ
Mesh
MeshCellCentroid
MeshCellCount
MeshCellHighlight
MeshCellIndex
MeshCellLabel
MeshCellMarker
MeshCellMeasure
MeshCellQuality
MeshCellShapeFunction
MeshCellStyle
MeshCells
MeshConnectivityGraph
MeshCoordinates
MeshFunctions
MeshPrimitives
MeshQualityGoal
MeshRefinementFunction
MeshRegion
MeshRegionQ
MeshShading
MeshStyle
Message
MessageList
MessageName
Messages
MexicanHatWavelet
MeyerWavelet
Midpoint
Midpoint3D
Min
MinColorDistance
MinDate
MinDetect
MinFilter
MinLimit
MinMax
MinStableDistribution
MinValue
MinimalBy
MinimalPolynomial
Minimize
MinimumSpanningTree
MinimumTimeIncrement
MinkowskiQuestionMark
Minors
Minus
MinusPlus
MissingBehavior
MissingDataMethod
MissingDataRules
MissingFallback
MissingQ
MissingValuePattern
MittagLefflerE
MixedFractionParts
MixedGraphQ
MixedRadix
MixtureDistribution
Mod
Mode
ModelFit
ModelFitReport
ModularInverse
ModularLambda
MoebiusMu
MolarVolumeIdealGas
Moment
MomentConvert
MomentEvaluate
MomentGeneratingFunction
MomentOfInertia
Momentum
Monitor
MonomialList
MonsterGroupM
MonteCarloIntegration
MorletWavelet
MorphologicalBinarize
MorphologicalBranchPoints
MorphologicalComponents
MorphologicalEulerNumber
MorphologicalGraph
MorphologicalPerimeter
MorphologicalTransform
MovingAverage
MovingMap
MovingMedian
MoyalDistribution
MultiaxisArrangement
Multicolumn
MultigraphQ
Multinomial
MultinomialDistribution
MultinormalDistribution
MultipleHarmonicNumber
MultiplePolyLog
MultipleZeta
MultiplicativeOrder
MultiplySides
MultivariateHypergeometricDistribution
MultivariatePoissonDistribution
MultivariateTDistribution

N

NArgMax
NArgMin
NBodySimulation
NCache
NCaputoD
NContourIntegrate
NDEigensystem
NDEigenvalues
NDSolve
NDSolveValue
NExpectation
NFractionalD
NIntegrate
NLineIntegrate
NMaxValue
NMaximize
NMinValue
NMinimize
NPV
NProbability
NProduct
NRoots
NSolve
NSolveValues
NSum
NSurfaceIntegrate
NakagamiDistribution
NameQ
Names
Nand
Nearest
NearestFunction
NearestMeshCells
NearestModel
NearestNeighborG
NearestNeighborGraph
NearestTo
NeedlemanWunschSimilarity
Needs
NegativeBinomialDistribution
NegativeDefiniteMatrixQ
NegativeMultinomialDistribution
NegativeSemidefiniteMatrixQ
NegativelyOrientedPoints
NeighborhoodGraph
NestGraph
NestTree
NestWhile
NestWhileList
NestedGreaterGreater

NestedLessLess
NeutronMass
NevilleThetaC
NevilleThetaD
NevilleThetaN
NevilleThetaS
NewtonRaphson
NextPrime
NextValue
NeymanScottPointProcess
NicholsPlot
Nominal
NominalScale
NominalVariables
NonCommutativeAlgebra
NonCommutativeCollect
NonCommutativeExpand
NonCommutativeGroebnerBasis
NonCommutativeMonomialList
NonCommutativeMultiply
NonCommutativePolynomialQ
NonCommutativePolynomialReduce
NonCommutativePolynomialReduction
NonCommutativeVariables
NoncentralBetaDistribution
NoncentralChiSquareDistribution
NoncentralFRatioDistribution
NoncentralStudentTDistribution
NoneMatch
NoneTrue
NonlinearModelFit
NonlocalMeansFilter
Nor
NorlundB
Norm
NormalDistribution
NormalMatrixQ
Normalize
NormalizedSquaredEuclideanDistance
NormalsFunction
NotCongruent
NotCupCap
NotDoubleVerticalBar
NotElement
NotEqualTilde
NotExists
NotGreater
NotGreaterEqual
NotGreaterFullEqual
NotGreaterGreater
NotGreaterLess
NotGreaterSlantEqual
NotGreaterTilde
NotHumpDownHump
NotHumpEqual
NotLeftTriangle
NotLeftTriangleBar
NotLeftTriangleEqual
NotLess
NotLessEqual
NotLessFullEqual
NotLessGreater
NotLessLess
NotLessSlantEqual
NotLessTilde
NotNestedGreaterGreater
NotNestedLessLess
NotPrecedes
NotPrecedesEqual
NotPrecedesSlantEqual
NotPrecedesTilde
NotReverseElement
NotRightTriangle
NotRightTriangleBar
NotRightTriangleEqual
NotSquareSubset
NotSquareSubsetEqual
NotSquareSuperset
NotSquareSupersetEqual
NotSubset
NotSubsetEqual
NotSucceeds
NotSucceedsEqual
NotSucceedsSlantEqual
NotSucceedsTilde
NotSuperset
NotSupersetEqual
NotTilde
NotTildeEqual
NotTildeFullEqual
NotTildeTilde
NotVerticalBar
NthDerivative
NullRawPointerQ
NullSpace
NumberCompose
NumberDecompose
NumberDigit
NumberExpand
NumberFieldClassNumber
NumberFieldDiscriminant
NumberFieldFundamentalUnits
NumberFieldIntegralBasis
NumberFieldNormRepresentatives
NumberFieldRegulator
NumberFieldRootsOfUnity
NumberFieldSignature
NumberForm
NumberLinePlot
Numerator
NumeratorDenominator
NumericArray
NumericArrayQ
NumericArrayType
NumericalOrder
NumericalSort
NyquistPlot

O

O
ODEClassify
ONanGroupON
Octahedron
Off
Offset
On
Opacity
OpacityFunction
OpacityFunctionScaling
OpaqueRawPointer
OpenAppend
OpenRead
OpenWrite
Opening
Operate
OperationDeclaration
OperatorApplied
OptionValue
Options
OrbitalVelocity
Order
OrderDistribution
OrderedQ
OrderedSchurDecomposition
Ordering
OrderingBy
Ordinal
OrdinalScale
OrnsteinUhlenbeckProcess
OrthogonalMatrix
OrthogonalMatrixQ
Orthogonalize
Out
OuterPolygon
OuterPolyhedron
OutputForm
Overflow
Overlay
OwenT

P

PCA
PDF
PERTDistribution
PMT
PackageExported
PackageImport
PackageInitialize
PackageScoped
PackedArrayQ
PadLeft
PadRight
PaddedForm
PadeApproximant
PageRank
PageRankCentrality
PairCorrelationG
PairedBarChart
PairedHistogram
PairedSmoothHistogram
PairedTTest
PairedZTest
PairwiseDensityHistogram
PairwiseListPlot
PairwiseProbabilityPlot
PairwiseQuantilePlot
PairwiseSmoothDensityHistogram
PalindromeQ
ParabolicCylinderD
ParallelArray
ParallelAxisPlot
ParallelCases
ParallelCombine
ParallelDo
ParallelEvaluate
ParallelKernels
ParallelMap
ParallelNeeds
ParallelProduct
ParallelSelect
ParallelSubmit
ParallelSum
ParallelTable

ParallelTry
Parallelepiped
Parallelization
Parallelize
Parallelogram
ParameterEstimator
ParameterMixtureDistribution
ParametricConvexOptimization
ParametricNDSolve
ParametricNDSolveValue
ParametricPlot
ParametricPlot3D
ParametricRegion
ParentDirectory
ParetoDistribution
ParetoPickandsDistribution
Parsec
PartialCorrelationFunction
PartialFractionElements
PartialFractions
Partition
PartitionCount
PartitionsList
PartitionsP
PartitionsQ
PascalBinomial
PascalDistribution
PathGraph
PathGraphQ
PathLength
PatternFilling
PaulWavelet
PauliMatrix
Pause
PeanoCurve
PearsonChiSquareTest
PearsonCorrelation
PearsonCorrelationTest
PearsonDistribution
PenttinenPointProcess
PercentForm
Percentile
PerfectNumber
PerfectNumberQ
PerformanceGoal
Perimeter
PeriodicModel
Periodogram
PeriodogramArray
Permanent
PermutationCycles
PermutationCyclesQ
PermutationGroup
PermutationLength
PermutationList
PermutationListQ
PermutationMatrix
PermutationMax
PermutationMin
PermutationOrder
PermutationPower
PermutationProduct
PermutationReplace
PermutationSupport
Permutations
Permute
PeronaMalikFilter
PerpendicularBisector
PerpetuityPV
PersistenceLocation
PetersenGraph
PfaffianDet
PhasePlane
PhongShading
PhotonEnergy
PhysicalConstant
Pick
PieChart
PieChart3D
Piecewise
PiecewiseExpand
PillaiTrace
PillaiTraceTest
PivotFromColumns
PivotTable
PivotToColumns
PixelValue
PixelValuePositions
Placed
PlanarAngle
PlanarFaceList
PlanarGraph
PlanarGraphQ
PlanckConstant
PlaneEquation
Plot
Plot2D
Plot3D
PlotFit
PlotFitElements
PlotGrid
PlotLabel
PlotLabels
PlotLegends
PlotMarkers
PlotPoints
PlotRange
PlotRangeClipping
PlotRangePadding
PlotRegion
PlotStyle
PlotTheme
PlusMinus
Pochhammer
Point
PointCountDistribution
PointDensity
PointDensityFunction
PointLegend
PointLight
PointProcessEstimator
PointProcessFitTest
PointProcessParameterAssumptions
PointProcessParameterQ
PointSize
PointStatisticFunction
PointToPlaneDistance
PointValuePlot
PoissonConsulDistribution
PoissonDistribution
PoissonPDEComponent
PoissonPointProcess
PoissonProcess
PolarAxes
PolarAxesOrigin
PolarCurve
PolarDecomposition
PolarGridLines
PolarPlot
PolarTicks
PolyGamma
PolyLog
PolyaAeppliDistribution
Polygon
PolygonAngle
PolygonArea
PolygonCoordinates
PolygonDecomposition
PolygonalNumber
Polyhedron
PolyhedronAngle
PolyhedronCoordinates
PolyhedronDecomposition
PolyhedronGenus
PolynomialDivision
PolynomialExpressionQ
PolynomialExtendedGCD
PolynomialGCD
PolynomialHermiteDecomposition
PolynomialHermiteReduce
PolynomialLCM
PolynomialMod
PolynomialModel
PolynomialQ
PolynomialQuotient
PolynomialQuotientRemainder
PolynomialReduce
PolynomialReduction
PolynomialRemainder
PolynomialSmithDecomposition
PolynomialSmithReduce
PolynomialSumOfSquaresList
PopovDecomposition
PositionIndex
PositionLargest
PositionSmallest
PositiveDefiniteMatrixQ
PositiveSemidefiniteMatrixQ
PositivelyOrientedPoints
PossibleZeroQ
Postfix
PotentialEnergy
PowerDistribution
PowerElectric
PowerExpand
PowerMod
PowerModList
PowerModel
PowerRange
PowerSet
PowerSpectralDensity
PowerSymmetricPolynomial
PowersRepresentations
Precedes
PrecedesEqual
PrecedesSlantEqual
PrecedesTilde
Precision
Prefix
PrependTo
PresentValue

PreserveColor
PriceGraphDistribution
Prime
PrimeFactorization
PrimeNu
PrimeOmega
PrimePi
PrimePowerQ
PrimeZetaP
PrimitivePolynomialQ
PrimitiveRoot
PrimitiveRootList
PrimitiveRootQ
PrincipalComponents
Print
PrintTemporary
PrintableASCIIQ
Prism
PrismVolume
PrivateKey
Probability
ProbabilityDistribution
ProbabilityPlot
ProbabilityScalePlot
ProbitModelFit
ProcessConnection
ProcessEstimator
ProcessInformation
ProcessParameterAssumptions
ProcessParameterQ
ProcessStatus
Processes
Product
ProductDistribution
ProductLog
ProjectileMaxHeight
ProjectileRange
Projection
Prolog
ProofObject
PropagateAborts
Proportion
Proportional
ProtonMass
Pruning
PseudoInverse
PublicKey
Put
PutAppend
Pyramid
PyramidVolume

Q

QBinomial
QFactorial
QGamma
QHypergeometricPFQ
QPochhammer
QPolyGamma
QRDecomposition
QnDispersion
QuadraticFormula
QuadraticIrrationalQ
QuadraticOptimization
QuadraticResidues
Quantile
QuantilePlot
Quartics
QuartileDeviation
QuartileSkewness
Quartiles
Query
QueueProperties
QueueingNetworkProcess
QueueingProcess
Quiet
QuietEcho
Quit
Quotient
QuotientRemainder

R

RFixedPoints
RGBColor
ROT13
RSolve
RSolveValue
RStabilityConditions
RadialAxisPlot
RadialGradientFilling
RadialGradientImage
RadialityCentrality
Radical
RadonTransform
RamanujanTau
RamanujanTauL
RamanujanTauTheta
RamanujanTauZ
Ramp
RandomChoice
RandomColor
RandomComplex
RandomDate
RandomFunction
RandomGraph
RandomImage
RandomInstance
RandomInteger
RandomPermutation
RandomPoint
RandomPointConfiguration
RandomPolygon
RandomPolyhedron
RandomPrime
RandomReal
RandomSample
RandomTime
RandomTree
RandomVariate
RandomWalkProcess
RangeFilter
RangeSpace
Rank
RankDecomposition
RankedMax
RankedMin
RarerProbability
Raster
Raster3D
Rasterize
RationalExpressionQ
RationalReconstruct
Rationalize
RationalizeDenominator
Ratios
RawMemoryAllocate
RawMemoryExport
RawMemoryFree
RawMemoryImport
RawMemoryRead
RawMemoryWrite
RawPointer
RayleighDistribution
Re
ReIm
ReImLabels
ReImPlot
ReImStyle
ReactionPDETerm
Read
ReadByteArray
ReadLine
ReadList
ReadString
RealAbs
RealDigits
RealExponent
RealPart
RealSign
RealValuedNumberQ
RealValuedNumericQ
Reap
ReconstructionMesh
Rectangle
RectangleChart
RectangleChart3D
RecurrenceTable
Reduce
ReducedPlanckConstant
ReferenceLineStyle
Refine
ReflectionMatrix
ReflectionTransform
Region
RegionBinarize
RegionBoundary
RegionBoundaryStyle
RegionBounds
RegionCentroid
RegionCongruent
RegionConvert
RegionDifference
RegionDilation
RegionDimension
RegionDisjoint
RegionDistance
RegionDistanceFunction
RegionEmbeddingDimension
RegionEqual
RegionErosion
RegionFarthestDistance
RegionFillingStyle
RegionFit
RegionFunction
RegionGaussianCurvature
RegionHausdorffDistance
RegionImage
RegionIntersection
RegionMaxCurvature
RegionMeanCurvature
RegionMeasure
RegionMember
RegionMemberFunction

RegionMinCurvature
RegionMoment
RegionNearest
RegionNearestFunction
RegionPlot
RegionPlot3D
RegionProduct
RegionQ
RegionResize
RegionSimilar
RegionSize
RegionSymmetricDifference
RegionUnion
RegionWithin
RegisterExceptionType
RegularPolygon
RegularPolygonArea
RegularlySampledQ
RelationGraph
RelationalDatabase
RelativisticMass
ReliabilityDistribution
ReliefImage
ReliefPlot
RemainingBalance
Remesh
RemoteFile
RemoveAlphaChannel
RemoveDiacritics
RemoveInputStreamMethod
RemoveOutputStreamMethod
RenameColumns
RenameDirectory
RenameFile
RenderingOptions
RenewalProcess
RenkoChart
RepairMesh
RepeatedTiming
RepeatingElement
Replace
ReplaceAt
ReplaceImageValue
ReplaceList
ReplacePart
ReplacePixelValue
Resampling
ResamplingMethod
Rescale
RescalingTransform
ResetDirectory
Residue
ResidueLabels
ResidueSum
Resolve
ResonanceFrequencyPendulum
ResonanceFrequencySpring
Resultant
ReverseApplied
ReverseBiorthogonalSplineWavelet
ReverseElement
ReverseEquilibrium
ReverseGraph
ReverseSort
ReverseSortBy
ReverseUpEquilibrium
RevolutionAxis
RevolutionPlot3D
RiccatiSolve
RiceDistribution
RidgeFilter
RiemannR
RiemannSiegelTheta
RiemannSiegelZ
RiemannXi
Riffle
RightArrow
RightArrowBar
RightArrowLeftArrow
RightCosetRepresentative
RightDownTeeVector
RightDownVector
RightDownVectorBar
RightTee
RightTeeArrow
RightTeeVector
RightTriangle
RightTriangleBar
RightTriangleEqual
RightUpDownVector
RightUpTeeVector
RightUpVector
RightUpVectorBar
RightVector
RightVectorBar
RipleyK
RipleyRassonRegion
RobustConvexOptimization
RogersTanimotoDissimilarity
RollPitchYawAngles
RollPitchYawMatrix
RomanNumeral
Root
RootApproximant
RootIntervals
RootMeanSquare
RootOfUnityQ
RootReduce
RootSum
RootTree
Roots
Rotate
RotateLabel
RotateLeft
RotateRight
RotationAction
RotationMatrix
RotationTransform
Round
RoundingRadius
Row
RowKey
RowReduce
RowSpace
RudinShapiro
RudvalisGroupRu
RulePlot
RulerUnits
RulesTree
Run
RunProcess
RunThrough
RungeKutta4
RuntimeAttributes
RuntimeOptions
RussellRaoDissimilarity
RydbergConstant
RydbergFormula

S

SARIMAProcess
SARMAProcess
SASTriangle
SATSolve
SHA1
SHA256
SHA512
SSSTriangle
SVD
SameTest
SatisfiabilityCount
SatisfiabilityInstances
SatisfiableQ
Save
SavitzkyGolayMatrix
SawtoothWave
ScalarTripleProduct
Scale
ScaleDivisions
ScaleOrigin
ScalePadding
ScaleRangeStyle
ScaleRanges
Scaled
ScalingFunctions
ScalingMatrix
ScalingTransform
Scan
ScatterPlot3D
SchrodingerPDEComponent
SchurDecomposition
ScientificForm
ScorerGi
ScorerGiPrime
ScorerHi
ScorerHiPrime
SearchIndices
Sec
SecantMethod
Sech
SechDistribution
SecondOrderConeOptimization
SectorChart
SectorChart3D
SectorOrigin
SectorSpacing
SecurityCertificate
SeedRandom
SelectComponents
SelectFirst
SemialgebraicComponentInstances
SemidefiniteOptimization
SeparableODE
SequenceAlignment
SequenceCases
SequenceCount
SequenceFold
SequenceFoldList
SequencePosition
SequenceReplace
SequenceSplit
SequenceType

SeriesCoefficient
SessionSubmit
SessionTime
SetAccuracy
SetAlphaChannel
SetCookies
SetDifference
SetDirectory
SetEnvironment
SetFileDate
SetFileFormatProperties
SetOptions
SetPrecision
SetSharedFunction
SetSharedVariable
SetStreamPosition
SetSystemOptions
ShannonWavelet
ShapiroWilk
ShapiroWilkTest
Share
Sharpen
ShearingMatrix
ShearingTransform
ShellRegion
ShenCastanMatrix
ShiftRegisterSequence
ShiftedGompertzDistribution
Short
ShortDownArrow
ShortLeftArrow
ShortRightArrow
ShortTimeFourier
ShortTimeFourierData
ShortUpArrow
ShortestCurveDistance
ShortestPath
ShortestPathFunction
Show
SiderealTime
SiegelTheta
SiegelTukeyTest
SierpinskiCurve
SierpinskiMesh
Sign
SignTest
Signature
SignedRankTest
SignedRegionDistance
SignificanceLevel
SimpleGraph
SimpleGraphQ
SimplePolygonQ
SimplePolyhedronQ
Simplex
Simplify
SimplifyMesh
SimpsonsRule
Sin
SinIntegral
Sinc
SinghMaddalaDistribution
SingularValueDecomposition
SingularValueList
SingularValuePlot
Sinh
SinhIntegral
SixJSymbol
Skeleton
SkeletonTransform
SkellamDistribution
SkewNormalDistribution
Skewness
Skip
SliceContourPlot3D
SliceDensityPlot3D
SliceDistribution
SliceVectorPlot3D
SlidingWindow
Slope
SmallCircle
SmithDecomposition
SmithReduce
SmithWatermanSimilarity
SmoothDateHistogram
SmoothDensityHistogram
SmoothHistogram
SmoothHistogram3D
SmoothKernelDistribution
SmoothMesh
SmoothPointDensity
SnDispersion
SnellsLaw
Snippet
SnubPolyhedron
SocketConnect
SocketListen
SocketListener
SocketObject
SocketOpen
SocketReadMessage
SocketReadyQ
SocketWaitAll
SocketWaitNext
SocketWriteMessage
Sockets
SokalSneathDissimilarity
SolarMass
SolidAngle
SolidBoundaryLoadValue
SolidDisplacementCondition
SolidFixedCondition
SolidMechanicsPDEComponent
SolidMechanicsStrain
SolidMechanicsStress
SolidRegionQ
SolveAlways
SolveValues
SortBy
SourcePDETerm
Sow
SpanningTree
SparseArray
SpatialBoundaryCorrection
SpatialEstimate
SpatialEstimatorFunction
SpatialGraphDistribution
SpatialJ
SpatialMedian
SpatialNoiseLevel
SpatialObservationRegionQ
SpatialPointData
SpatialPointSelect
SpatialRandomnessTest
SpatialTrendFunction
SpearmanCorrelation
SpearmanRankTest
SpearmanRho
SpectralNorm
Spectrogram
SpectrogramArray
Specularity
SpeedOfLight
Sphere
SpherePoints
SphereSurfaceArea
SphereVolume
SphericalAngle
SphericalBesselJ
SphericalBesselY
SphericalDistance
SphericalHankelH1
SphericalHankelH2
SphericalHarmonicY
SphericalPlot3D
SphericalRegion
SphericalShell
SpheroidalEigenvalue
SpheroidalJoiningFactor
SpheroidalPS
SpheroidalPSPrime
SpheroidalQS
SpheroidalQSPrime
SpheroidalRadialFactor
SpheroidalS1
SpheroidalS1Prime
SpheroidalS2
SpheroidalS2Prime
SplicedDistribution
SplineClosed
SplineDegree
SplineKnots
SplineWeights
Split
SplitBy
SpotLight
Sqrt
Square
SquareFreeQ
SquareIntersection
SquareMatrixQ
SquareSubset
SquareSubsetEqual
SquareSuperset
SquareSupersetEqual
SquareUnion
SquareWave
SquaredEuclideanDistance
SquaresR
StableDistribution
StackedDateListPlot
StackedListPlot
StadiumShape
StandardAtmosphere
StandardDeviation
StandardDeviationFilter
StandardError
StandardForm
StandardGravity
Standardize
StandbyDistribution

Star
StarGraph
StartProcess
StateSpaceModel
StationaryDistribution
StationaryWaveletPacketTransform
StationaryWaveletTransform
StefanBoltzmann
StefanBoltzmannConstant
StieltjesGamma
StippleShading
Stirling
StirlingS1
StirlingS2
StrataVariables
StratonovichProcess
StraussHardcorePointProcess
StraussPointProcess
StreamColorFunction
StreamColorFunctionScaling
StreamDensityPlot
StreamMarkers
StreamPlot
StreamPlot3D
StreamPoints
StreamPosition
StreamScale
StreamStyle
Streams
StrictInequalities
StringApply
StringCapitalize
StringCases
StringCenter
StringContainsQ
StringCount
StringDelete
StringDrop
StringEndsQ
StringExtract
StringForm
StringFormat
StringFormatQ
StringFreeQ
StringInsert
StringMatchQ
StringPadLeft
StringPadRight
StringPart
StringPartition
StringPosition
StringRepeat
StringReplace
StringReplaceList
StringReplacePart
StringReverse
StringRiffle
StringRotateLeft
StringRotateRight
StringSkeleton
StringSplit
StringStartsQ
StringTake
StringTakeDrop
StringTemplate
StringToByteArray
StringToLowerCase
StringToStream
StringToTitleCase
StringToUpperCase
StringTrim
StruveH
StruveL
StudentTDistribution
Style
Subdivide
SubdivisionRegion
Subfactorial
Subgraph
SubresultantPolynomialRemainders
SubresultantPolynomials
Subresultants
Subsequences
Subset
SubsetCases
SubsetCount
SubsetEqual
SubsetMap
SubsetPosition
SubsetQ
SubsetReplace
Subsets
SubstitutionSystem
SubtractSides
Succeeds
SucceedsEqual
SucceedsSlantEqual
SucceedsTilde
Success
SuchThat
Sum
SumConvergence
Superset
SupersetEqual
Surd
SurfaceArea
SurfaceContourPlot3D
SurfaceDensityPlot3D
SurfaceIntegrate
SurvivalDistribution
SurvivalFunction
SurvivalModel
SurvivalModelFit
SuzukiDistribution
SuzukiGroupSuz
SwatchLegend
Symbol
SymbolName
SymbolicDeltaProductArray
SymbolicIdentityArray
SymbolicOnesArray
SymbolicZerosArray
SymletWavelet
Symmetric
SymmetricDifference
SymmetricGroup
SymmetricKey
SymmetricMatrix
SymmetricMatrixQ
SymmetricPolynomial
SymmetricReduction
Symmetrize
SymmetrizedArray
SymmetrizedArrayRules
SymmetrizedDependentComponents
SymmetrizedIndependentComponents
SymmetrizedReplacePart
SyntaxLength
SyntaxQ
SystemInformation
SystemOpen
SystemOptions
SystemProcesses

T

TDistribution
TTest
TableForm
Tabular
TabularColumn
TabularColumnQ
TabularQ
TabularRow
TabularRowQ
TabularSchema
TabularStructure
TabularSummary
TaggedNestGraph
TakeDrop
TakeLargest
TakeLargestBy
TakeList
TakeSmallest
TakeSmallestBy
TakeWhile
Tally
Tan
Tanh
TaskAbort
TaskExecute
TaskRemove
TaskResume
TaskSuspend
TaskWait
Tasks
TautologyQ
TaylorPolynomial
TaylorSeries
TeXForm
TelegraphProcess
TemplateApply
TemporalRegularity
TensorContract
TensorDimensions
TensorExpand
TensorProduct
TensorRank
TensorReduce
TensorSymmetry
TensorTranspose
TensorWedge
TerminatedEvaluation
TernaryListPlot
TernaryPlotCorners
TestCreate
TestEvaluate
TestEvaluationFunction
TestID
TestObject
TestReport
TestReportObject
Tetrahedron

TetrahedronVolume
Text
TextElement
TextSearch
TextSearchReport
TextString
TextWords
Texture
TextureCoordinateFunction
TextureCoordinateScaling
TextureMapping
Therefore
ThermalExpansion
ThermometerGauge
Thickness
Thinning
ThomasPointProcess
ThompsonGroupTh
ThreeJSymbol
Threshold
Through
ThrowException
ThueMorse
Thumbnail
TickDirection
TickLabelOrientation
TickLabelPositioning
TickLabels
TickLengths
TickPositions
Ticks
TicksStyle
Tilde
TildeEqual
TildeFullEqual
TildeTilde
TimeConstrained
TimeConstraint
TimeDilation
TimeObject
TimeObjectQ
TimeObservationWindow
TimeRemaining
TimeSeries
TimeSeriesAggregate
TimeSeriesEvents
TimeSeriesForecast
TimeSeriesInsert
TimeSeriesInvertibility
TimeSeriesMap
TimeSeriesMapThread
TimeSeriesModel
TimeSeriesModelFit
TimeSeriesQ
TimeSeriesResample
TimeSeriesRescale
TimeSeriesShift
TimeSeriesStructure
TimeSeriesSummary
TimeSeriesThread
TimeSeriesWindow
TimeString
TimeSystemConvert
TimeUsed
TimelinePlot
Timing
TitsGroupT
ToCharacterCode
ToExpression
ToFiniteField
ToInvertibleTimeSeries
ToLowerCase
ToMemory
ToNumberField
ToPackedArray
ToPolarCoordinates
ToRadicals
ToRawPointer
ToRules
ToSphericalCoordinates
ToTabular
ToUpperCase
ToeplitzMatrix
ToonShading
TopHatTransform
TopologicalSort
Torus
TorusGraph
TorusSurfaceArea
TorusVolume
TotalVariationFilter
Tr
Trace
TracePrint
TraceScan
TracyWidomDistribution
TradingChart
TraditionalForm
TransferFunctionModel
TransformColumns
TransformMissing
TransformationFunction
TransformationFunctions
TransformationMatrix
TransformedDistribution
TransformedField
TransformedProcess
TransformedRegion
TransitiveClosureGraph
TransitiveReductionGraph
Translate
TranslationTransform
TransposeList
TrapezoidalRule
Tree
TreeCases
TreeChildren
TreeCount
TreeData
TreeDelete
TreeDepth
TreeExpression
TreeExtract
TreeFold
TreeForm
TreeGame
TreeGamePayoff
TreeGamePlot
TreeGraph
TreeGraphQ
TreeInsert
TreeLeafCount
TreeLeafQ
TreeLeaves
TreeLevel
TreeMap
TreeMapAt
TreePlot
TreePosition
TreeQ
TreeReplacePart
TreeRules
TreeScan
TreeSelect
TreeSize
TriGamma
Triangle
TriangleArea
TriangleCenter
TriangleConstruct
TriangleMeasurement
TriangleWave
TriangularDistribution
TriangulateMesh
Trig
TrigExpand
TrigFactor
TrigFactorList
TrigReduce
TrigToExp
TrimmedMean
TrimmedVariance
TruncateSum
TruncatedDistribution
TruncatedPolyhedron
TruthTable
TsallisQExponentialDistribution
TsallisQGaussianDistribution
Tube
TukeyLambdaDistribution
Tuples
TuranGraph
TuringMachine
TuttePolynomial
TwoWayRule
TypeCheck
TypeDeclaration
TypeEvaluate
TypeHint
TypeMatchQ
TypeOf
TypeSignature
TypeSpecifier
Typed

U

URL
URLBuild
URLDecode
URLDownload
URLDownloadSubmit
URLEncode
URLExecute
URLExpand
URLParse
URLQueryDecode
URLQueryEncode
URLRead
URLResponseTime
URLSubmit
UnateQ
UncertaintyProduct

Uncompress
Underflow
UndirectedEdge
UndirectedGraph
UndirectedGraphQ
UniformDistribution
UniformGraphDistribution
UniformPolyhedron
UniformSumDistribution
UnilateralConvolve
UnilateralDiscreteConvolve
Union
UnionPlus
UniqueElements
UnitRootTest
UnitStep
UnitTriangle
UnitVector
UnitaryMatrix
UnitaryMatrixQ
Unitize
UnixTime
UnlabeledTree
UnmanageObject
UnsetShared
UpArrow
UpArrowBar
UpArrowDownArrow
UpDownArrow
UpEquilibrium
UpTee
UpTeeArrow
Update
UpdateSearchIndex
UpperCaseQ
UpperLeftArrow
UpperRightArrow
UpperTriangularMatrix
UpperTriangularMatrixQ
UpperTriangularize

V

VacuumPermeability
VacuumPermittivity
Values
VandermondeMatrix
Variables
Variance
VarianceEquivalenceTest
VarianceEstimatorFunction
VarianceGammaDistribution
VarianceGammaPointProcess
VarianceTest
VariogramFunction
VariogramModel
VectorAngle
VectorAround
VectorAspectRatio
VectorColorFunction
VectorColorFunctionScaling
VectorDensityPlot
VectorDisplacementPlot
VectorDisplacementPlot3D
VectorFieldPlot
VectorGreater
VectorGreaterEqual
VectorLess
VectorLessEqual
VectorMarkers
VectorPlot
VectorPlot3D
VectorPoints
VectorQ
VectorRange
VectorScaling
VectorSizes
VectorStyle
VectorSymbol
Vectors
Vee
Velocity
VerificationTest
Verify
VerifyDerivedKey
VerifyDigitalSignature
VerifyFileSignature
VerifyMatrixGameStrategy
VerifySolutions
VerifyTestAssumptions
VerifyTreeGameStrategy
VertexAdd
VertexChromaticNumber
VertexColors
VertexComponent
VertexConnectivity
VertexContract
VertexCorrelationSimilarity
VertexCosineSimilarity
VertexCount
VertexCoverQ
VertexDataCoordinates
VertexDegree
VertexDelete
VertexDiceSimilarity
VertexEccentricity
VertexInComponent
VertexInComponentGraph
VertexInDegree
VertexIndex
VertexJaccardSimilarity
VertexList
VertexNormals
VertexOutComponent
VertexOutComponentGraph
VertexOutDegree
VertexQ
VertexReplace
VertexTextureCoordinates
VertexTransitiveGraphQ
VertexWeight
VertexWeightedGraphQ
VerticalBar
VerticalGauge
VerticalSeparator
VerticalTilde
ViewAngle
ViewCenter
ViewMatrix
ViewPoint
ViewProjection
ViewRange
ViewVector
ViewVertical
VoigtDistribution
Volume
VonMisesDistribution
VonMisesStress
VoronoiMesh

W

WaitAll
WaitNext
WakebyDistribution
WalleniusHypergeometricDistribution
WaringYuleDistribution
WarpingCorrespondence
WarpingDistance
WatershedComponents
WatsonUSquareTest
WattsStrogatzGraphDistribution
WavePDEComponent
WaveSpeed
WaveletBestBasis
WaveletFilterCoefficients
WaveletImagePlot
WaveletListPlot
WaveletMapIndexed
WaveletMatrixPlot
WaveletPhi
WaveletPsi
WaveletScalogram
WaveletThreshold
WaveletTransform
WeakStationarity
WeaklyConnectedComponents
WeaklyConnectedGraphComponents
WeaklyConnectedGraphQ
WeberE
Wedge
WeibullDistribution
WeierstrassE1
WeierstrassE2
WeierstrassE3
WeierstrassEta1
WeierstrassEta2
WeierstrassEta3
WeierstrassHalfPeriodW1
WeierstrassHalfPeriodW2
WeierstrassHalfPeriodW3
WeierstrassHalfPeriods
WeierstrassInvariantG2
WeierstrassInvariantG3
WeierstrassInvariants
WeierstrassP
WeierstrassPPrime
WeierstrassSigma
WeierstrassZeta
WeightedAdjacencyGraph
WeightedAdjacencyMatrix
WeightedGraphQ
Weights
WeylAlgebra
WheelGraph
WhiteNoiseProcess
WhitePoint
WhittakerM
WhittakerW
WienDisplacement
WienDisplacementConstant
WienerFilter
WienerProcess
WignerD
WignerSemicircleDistribution

WilcoxonTest
WilksW
WilksWTest
WindingCount
WindingPolygon
WinsorizedMean
WinsorizedVariance
WishartMatrixDistribution
WithCleanup
WithLock
WordCloud
WordCount
WordCounts
WordStem
Work
Write
WriteLine
WriteString
Wronskian

X

XMLTemplate
XORCipher
XYZColor
Xnor
Xor

Y

YoungTableaux
YuleDissimilarity

Z

ZScore
ZScoreNormalize
ZTest
ZTransform
ZernikeR
ZeroMatrix
ZeroSymmetric
ZeroTest
Zeta
ZetaZero
ZipfDistribution
ZoomCenter
ZoomFactor

11 Rebuilding and contributing

Follow `manual/REBUILD.md`. Regenerate the live function reference and execute the example checker before compiling the PDF. Do not edit generated files by hand. Keep source and PDF together so the document can be reproduced.

Contributions should test the actual public entry point and state unsupported inputs. For a bug, demonstrate a failing regression before the narrow repair. A name count is not a correctness score, and a passing example does not justify removing an honest limitation.

MikoshiLang is distributed under Apache License 2.0; consult `LICENSE`. It relies on the Python scientific ecosystem, especially SymPy and NumPy. Wolfram Language and Mathematica are separate products; no affiliation, endorsement or full compatibility is claimed.