Mojo🔥 modules

These are all the modules in the Mojo standard library.

anytype

Defines the `AnyType` trait

arc

Reference-counted smart pointers

arg

Implements functions and variables for interacting with execution and system environment

atomic

Implements the Atomic class

base64

Provides functions for base64 encoding strings

bencher

benchmark

Implements the benchmark module for runtime benchmarking

bit

Provides functions for bit manipulation

bool

Implements the Bool class

breakpoint

This module includes the builtin breakpoint function

buffer

builtin_list

Implements the Buffer class

builtin_slice

Implements slice

compiler

constants

Defines math utilities

coroutine

Implements classes and methods for coroutines

debug

This module includes the debug hook functions

dict

Defines `Dict`, a collection that stores key-value pairs

dtype

Implements the ListLiteral class

comparable

complex

Implements the Complex type

constrained

Implements compile time constraints

counter

Defines the `Counter` type

debug_assert

Implements a debug assert

dimlist

Provides utilities for working with static and variadic lists

env

Implements the DType class

equality_comparable

ffi

Implements a foreign functions interface (FFI)

file_descriptor

Higher level abstraction for file stream

format

Implements a formatter abstraction for objects that can format themselves to a string

fstat

Implements the file system stat operations

hash

Implements the `Hashable` trait and

index

Implements basic routines for working with the OS

error

Implements the Error class

file

Implements the file based methods

float_literal

Implements the FloatLiteral class

format_int

Provides the `hex` and

functional

Implements higher-order functions

identifiable

info

Implements `StaticIntTuple` which is commonly used to represent N-D indices

inline_array

Defines the `InlineArray` type

inline_string

Implements a string that has a small-string optimization which avoids heap allocations for short strings

int_literal

Implements the IntLiteral class

io

Provides utilities for working with input/output

list

Defines the List type

loop

Implements methods for querying the host target info

inline_list

Defines the `InlineList` type

int

Implements the Int class

intrinsics

Defines intrinsics

len

Provides the `len()` function and its associated traits

lock

math

Defines math utilities

math

Defines basic math functions for use in the open source parts of the standard library since the `math` package is currently closed source and cannot be depended on in the open source parts of the standard library

maybe_uninitialized

memory

Defines functions for memory manipulations

memory

memory

Implements `parallel_memcpy`

none

Defines the builtin `NoneType`

numerics

Defines utilities to work with numeric types

object

Defines the object type, which is used to represent untyped values

optional

Defines Optional, a type modeling a value which may or may not be present

os

Implements os methods

param_env

Implements functions for retrieving compile-time defines

path

Implements `Path` and related functions

path

pathlike

Implements PathLike trait

polynomial

Provides two implementations for evaluating polynomials

pwd

python

Implements Python interoperability

python_object

Implements PythonObject

quick_bench

random

Provides functions for random numbers

range

Implements a 'range' call

rebind

Implements type rebind

reduction

Implements SIMD reductions

reference

Implements the Reference type

repr

Provide the `repr` function

reversed

Provides the `reversed` function for reverse iteration over collections

set

Implements the Set datatype

simd

Implements SIMD struct

sort

Implements the built-in `sort` function

span

Implements the Span type

stat

Implements the stat module

static_tuple

Implements StaticTuple, a statically-sized uniform container

str

Provides the `str` function

string

Implements basic object methods for working with strings

string_literal

Implements the StringLiteral class

string_slice

Implements the StringSlice type

stringref

Implements the StringRef class

swap

Implements the built-in `swap` function

tempfile

Implements tempfile methods

terminate

This module includes the exit functions

testing

Implements various testing utils

time

Implements basic utils for working with time

tuple

Implements the Tuple type

type_aliases

Defines some type aliases

uint

Implements the UInt class

unsafe

Implements types that work with unsafe pointers

unsafe_pointer

Implement a generic unsafe pointer type

value

Defines core value traits

variant

Defines a Variant type

vector

Defines InlinedFixedVector