



Mojo traits cheat sheet

What each standard library trait does, when you need it, and where it bites.

v1.3dev

CONFORMING

Conformance syntax declare, condition, or opt out of a trait

```
struct Meters(Comparable, Writable): ...
    # unconditional conformance
struct Box[T: AnyType](Hashable where
    conforms_to(T, Hashable)): ...
    # only when T does, usually a field or fields
struct Handle(not Movable): ...
    # opt out: pinned in place
struct Lease(not Deinitable else "call
    'release()'): ...
    # opt out, with a reason: linear type
```

A conditional conformance exists when the compiler can prove a condition: `Box[Int]` is `Hashable`, and a `Box` of a non-hashable type isn't.

A `not Deinitable` message appears in the error when a value reaches the end of scope without calling its deinitializer.

DIAGNOSTICS

Messages after else

what to write when a condition or opt-out fails

```
def chunk[w: Int]() where w.is_power_of_two()
    else "'w' must be a power of two": pass

struct Lease(
    not Deinitable else "call 'release()' to return
    the lease"
): pass
```

BEST PRACTICES

A compilation error says what went wrong; your `else` message says what to do next. The compiler appends your message, so write it as a continuation: lowercase, with no final period.

WRITING ELSE MESSAGES

After `else`, state what's required: `'count' must be positive`.

After `not Deinitable`, state the method that ends the value: `call 'release()'`.

Put identifiers in single quotes, not backticks.

Don't restate the failure. The compiler already says what went wrong.

Use a string literal. A computed message doesn't compile.

USING THIS CHART

How to read this chart all the traits in one place

TRAIT DECLARATION INFORMATION

FEATURE	DESCRIPTION
source	Path to the stdlib file that defines the trait (top right)
implicit	Automatically applied by default if the type supports it (top left)
marker	Trait that <i>marks</i> a type without requirements (top left)
refines	An ordered list of traits that this trait builds on (after header) < is a refinement dependency Multiple names form trait compositions

TRAIT REQUIREMENTS

The methods or declarations required by the trait.

FEATURE	DESCRIPTION
provided	A default implementation is available
associated type	The type declared as a comptime member
compile-time flag	The flag declared as a comptime member
no remark	The type must implement it

Notes follow where applicable.

LIFECYCLE

AnyType (implicit, marker) the root; every type conforms

`traits/anytype.mojo`

No requirements (marker).

Deinitable (implicit) has a destructor, called at last use

traits/deinitable.mojo

SIGNATURE	REMARK
__deinit__(deinit self, /)	provided
comptime __del__is_trivial: Bool	compile-time flag

Automatically added to every eligible type (all fields are also Deinitable). When the type is trivial, Mojo skips the destructor. Opt out with not Deinitable to make a linear type.

Movable (implicit) relocate a value to new storage

traits/movable.mojo

SIGNATURE	REMARK
__init__(out self, *, deinit move: Self)	provided
comptime __move_ctor_is_trivial: Bool	compile-time flag

Every type is Movable unless it opts out. not Movable pins a value in place: it can't go to a new binding (var b = a^), out of a function (return local^), or into a container (append()). A ^ transfer that consumes the value where it is, into a var argument or a deinit method, still works.

Copyable explicit copy

traits/copyable.mojo

Refines: Movable

SIGNATURE	REMARK
__init__(out self, *, copy: Self)	provided
copy(self) -> Self	provided
comptime __copy_ctor_is_trivial: Bool	compile-time flag

The copy constructor is synthesized if all fields are Copyable. Call it with .copy().

ImplicitlyCopyable (marker) lets copies be inserted implicitly

traits/copyable.mojo

Refines: Copyable < Movable

No additional requirements.

Use thoughtfully, not to save time typing .copy(). Prefer Movable or Copyable when you don't need implicit copies for function arguments, constructors, or similar contexts.

Defaultable create with no arguments

builtin/value.mojo

SIGNATURE	REMARK
__init__(out self)	

Reach for this when you need parameterized default-construction (TypeName()) without arguments.

RegisterPassable (marker) stored in registers

builtin/value.mojo

STORED IN REGISTERS, NOT MEMORY; NO STABLE ADDRESS OR IDENTITY

Refines: Movable

No requirements (marker).

No stable address: you can't take the address of self in imm-convention methods. Identifiable is meaningless for these types.

Moved trivially; all fields must also conform.

TrivialRegisterPassable (marker) copyable by moving bits

builtin/value.mojo

REGISTER-PASSABLE, COPYABLE BY MOVING BITS, NO SIDE EFFECTS.

Refines: ImplicitlyCopyable < Copyable < Movable, Deinitable, RegisterPassable

No requirements (marker).

A type whose values are treated as basic bit patterns. No constructors or destructors needed. All fields must also conform.

FORMAT

Writable format itself as text

format/__init__.mojo

FORMAT ITSELF AS TEXT; WORKS WITH PRINT(), STRING(), FORMAT STRINGS

SIGNATURE	REMARK
write_to(self, mut writer: Some[Writer])	provided
write_repr_to(self, mut writer: Some[Writer])	provided

If all fields conform, you inherit both methods through reflection.

Writer custom output target

format/__init__.mojo

SIGNATURE	REMARK
write_string(mut self, string: StringSpan)	
write[*Ts: Writable](mut self, *args: *Ts)	provided

Use for loggers, network streams, string builders, etc. String, FileHandle, RawFileDescriptor conform to Writer.

TESTING

Strategy random inputs

testing/prop/strategy/__init__.mojo

RANDOM INPUTS FOR PROPERTY-BASED TESTS

Refines: Deinitable, Movable

SIGNATURE	REMARK
Value: Copyable & Deinitable	associated type
value(mut self, mut rng: Rng) raises -> Self.Value	

Allows strategies to carry and advance state between draws. value() draws one sample from the random number generator.

COMPARE & HASH

Equatable equality; enables == and !=

builtin/comparable.mojo

SIGNATURE	REMARK
__eq__(self, other: Self, /) -> Bool	provided
__ne__(self, other: Self, /) -> Bool	provided

Don't use with floating-point values (use isclose()).

NaN != NaN. (See: std.utils.numerics.nan)

Mojo provides a fieldwise default. Override for caches, internal metadata, and custom behavior.

Comparableordered comparison; < > ≤ ≥, and sort()
builtin/comparable.mojo

Refines: Equatable

SIGNATURE	REMARK
__lt__(self, rhs: Self) -> Bool	
__gt__(self, rhs: Self) -> Bool	provided
__le__(self, rhs: Self) -> Bool	provided
__ge__(self, rhs: Self) -> Bool	provided

Implement __lt__() unless it's expensive. If so, override all four.

Hashablea type that can be hashed
hash/hash.mojo

KeyElement = Hashable + Equatable + Movable

SIGNATURE	REMARK
__hash__(self, mut hasher: Some[Hasher])	provided

Needed for Dict keys and Set elements.

Hasherimplements a hash algorithm
hash/haser.mojo

SIGNATURE	REMARK
__init__(out self)	
__update_with_simd(mut self, value: SIMD[_,_])	
update(mut self, value: ImmSpan[Byte, _])	
finish(var self) -> UInt64	

Every method is required. update() takes bytes; a Hashable type feeds itself to a hasher through its own __hash__().

Identifiableidentity; same-object test, enables is / is not
builtin/identifiable.mojo

SIGNATURE	REMARK
__is__(self, rhs: Self) -> Bool	
__isnot__(self, rhs: Self) -> Bool	provided

Identity is meaningless for register-passable types, which have no stable address.

CONVERT

Boolable, Intable, Floatableconvert with Bool(), Int(), Float64()
builtin/bool.mojo · builtin/int.mojo · builtin/floatable.mojo

LOGICAL ("TRUTHY") AND NUMERIC CONVERSIONS

SIGNATURE	REMARK
__bool__(self) -> Bool	Boolable
__int__(self) -> Int	Intable
__int__(self) raises -> Int	IntableRaising
__float__(self) -> Float64	Floatable
__float__(self) raises -> Float64	FloatableRaising

Boolable unlocks use with if / while / and / or like if my_boolable_object: do_something().

Intable converts to any integer type; Floatable only to Float64. If the method can fail (Int("two")), use the Raising variant.

MATH

Absable, Powable, Roundableunary math operators
math/math.mojo

SIGNATURE	REMARK
__abs__(self) -> Self	Absable · abs()
__pow__(self, exp: Self) -> Self	Powable · pow(), **
__round__(self) -> Self	Roundable · round()
__round__(self, ndigits: Int) -> Self	Roundable · round(), precision

Ceilable, Floorable, Truncableround toward a bound
math/math.mojo

SIGNATURE	REMARK
__ceil__(self) -> Self	Ceilable · ceil()
__floor__(self) -> Self	Floorable · floor()
__trunc__(self) -> Self	Truncable · trunc()

CeilDivable, CeilDivableRaisingceiling division
math/math.mojo

SIGNATURE	REMARK
__ceildiv__(self, denominator: Self) -> Self	CeilDivable
__ceildiv__(self, denominator: Self) raises -> Self	CeilDivableRaising

Rounds up instead of down.

DivModablecombined division and modulo; enables divmod()
math/math.mojo

Refines: ImplicitlyCopyable < Copyable < Movable

SIGNATURE	REMARK
__divmod__(self, denominator: Self) -> Tuple[Self, Self]	DivModable

Math outlier. The tuple is (quotient, remainder).

COUNT AND ITERATE

Sized, SizedRaisinghas a length; enables len()
builtin/len.mojo

SIGNATURE	REMARK
__len__(self) -> Int	Sized
__len__(self) raises -> Int	SizedRaising

Don't use with string. String lengths depend on character encoding. Ask for byte_length(), count_codepoints(), or count_graphemes() and skip Sized.

Iterableiterate by borrowing
iter/__init__.mojo

SIGNATURE	REMARK
IteratorType[iterable_mut: Bool, //, iterable_origin: Origin[mut=iterable_mut]]: Iterator	associated type
__iter__(ref self) -> Self.IteratorType[origin_of(self)]	

The iterator yields references tied to the source lifetime. The origin is tracked.

IterableOwned iterate by consuming and owning

iter/__init__.mojo

SIGNATURE	REMARK
IteratorOwnedType: Iterator	associated type
<code>__iter__(var self) -> Self.IteratorOwnedType</code>	

No origin tracking.

Iterator

produces elements one at a time; the for-loop workhorse

iter/__init__.mojo

Refines: Deinitable, Movable

SIGNATURE	REMARK
Element: Movable	associated type
<code>__next__(mut self) raises StopIteration -> Self.Element</code>	
<code>bounds(self) -> Tuple[Int, Optional[Int]]</code>	provided
<code>nth(var self, n: Int) -> Optional[Self.Element]</code>	provided

Requires an Iterable on the collection, and Iterator on the iterator. Don't rely on bounds() for safety checks; it's a hint.

Typed raise: StopIteration.

INTEROP

PathLike treat as a conforming path

os/pathlike.mojo

Represents itself as a file system path.

SIGNATURE	REMARK
<code>__fspath__(self) -> String</code>	

ConvertibleToPython can be sent to Python

python/conversions.mojo

Refines: Deinitable

SIGNATURE	REMARK
<code>to_python_object(var self) raises -> PythonObject</code>	

ConvertibleFromPython can be created from a Python object

python/conversions.mojo

Refines: Copyable < Movable, Deinitable

SIGNATURE	REMARK
<code>__init__(out self, *, py: PythonObject) raises</code>	