Swift / Swift Standard Library / Basic Behaviors

API Collection

# Basic Behaviors

Use your custom types in operations that depend on testing for equality or order and as members of sets and dictionaries.

## Topics

### Equality and Ordering

Allow your custom types to be used with simple collection operations, such as `contains(_:)`, and standard comparison operators.

`protocol Equatable`

A type that can be compared for value equality.

`protocol Comparable`

A type that can be compared using the relational operators <, <=, >=, and >.

`protocol Identifiable`

A class of types whose instances hold the value of an entity with stable identity.

### Copying

`protocol Copyable`

A type whose values can be implicitly or explicitly copied.

`protocol BitwiseCopyable`

### Sets and Dictionaries

Store your custom types in sets or use them as dictionary keys.

`protocol Hashable`

A type that can be hashed into a `Hasher` to produce an integer hash value.

`struct Hasher`

The universal hash function used by `Set` and `Dictionary`.

## String Representation

`protocol CustomStringConvertible`

A type with a customized textual representation.

`protocol LosslessStringConvertible`

A type that can be represented as a string in a lossless, unambiguous way.

`protocol CustomDebugStringConvertible`

A type with a customized textual representation suitable for debugging purposes.

## Raw Representation

`protocol CaseIterable`

A type that provides a collection of all of its values.

`protocol RawRepresentable`

A type that can be converted to and from an associated raw value.

# See Also

## Tools for Your Types

Encoding, Decoding, and Serialization

Serialize and deserialize instances of your types with implicit or customized encoding.

Initialization with Literals

Allow values of your type to be expressed using different kinds of literals.