

[Foundation](#) / Collections

API Collection

# Collections

Use arrays, dictionaries, sets, and specialized collections to store and iterate groups of objects or values.

## Topics

### Basic Collections

`@frozen struct Array<Element>`

An ordered, random-access collection.

`@frozen struct Dictionary<Key, Value> where Key : Hashable`

A collection whose elements are key-value pairs.

`@frozen struct Set<Element> where Element : Hashable`

An unordered collection of unique elements.

### Indexes

`struct IndexPath`

A list of indexes that together represent the path to a specific location in a tree of nested arrays.

`struct IndexSet`

A collection of unique integer values that represent the indexes of elements in another collection.

### Specialized Sets

`class NSCountedSet`

A mutable, unordered collection of distinct objects that may appear more than once in the collection.

`class NSOrderedSet`

A static, ordered collection of unique objects.

`class NSMutableOrderedSet`

A dynamic, ordered collection of unique objects.

## Purgeable Collections

`class NSCache`

A mutable collection you use to temporarily store transient key-value pairs that are subject to eviction when resources are low.

`class NSPurgeableData`

A mutable data object containing bytes that can be discarded when they're no longer needed.

## Pointer Collections

`class NSPointerArray`

A collection similar to an array, but with a broader range of available memory semantics.

`class NSMapTable`

A collection similar to a dictionary, but with a broader range of available memory semantics.

`class NSHashTable`

A collection similar to a set, but with broader range of available memory semantics.

## Iteration

`class NSEnumerator`

An abstract class whose subclasses enumerate collections of objects, such as arrays and dictionaries.

`protocol NSFastEnumeration`

A protocol that objects adopt to support fast enumeration.

`struct NSFastEnumerationIterator`

```
struct NSIndexSetIterator
```

An iterator suitable for enumerating the elements of an index set.

```
struct NSEnumerationOptions
```

Options for block enumeration operations.

```
struct NSSortOptions
```

Options for block sorting operations.

## Special Semantic Values

```
class NSNull
```

A singleton object used to represent null values in collection objects that don't allow nil values.

```
let NSNotFound: Int
```

```
let NSNotFound: Int
```

A value indicating that a requested item couldn't be found or doesn't exist.

## See Also

### Fundamentals

☰ Numbers, Data, and Basic Values

Work with primitive values and other fundamental types used throughout Cocoa.

☰ Strings and Text

Create and process strings of Unicode characters, use regular expressions to find patterns, and perform natural language analysis of text.

☰ Dates and Times

Compare dates and times, and perform calendar and time zone calculations.

☰ Units and Measurement

Label numeric quantities with physical dimensions to allow locale-aware formatting and conversion between related units.

☰ Data Formatting

Convert numbers, dates, measurements, and other values to and from locale-aware string representations.

## ☰ Filters and Sorting

Use predicates, expressions, and sort descriptors to examine elements in collections and other services.