

Swift 4 Standard Library

Programming Tasks

Input & Output

Debugging & Reflection

Key-Path Expressions

Manual Memory Management

Type Casting and Existential Types

Perform casts between types or represent values of any type.

Integer Value Casting

```
func numericCast<T, U>(T) U
Returns the given integer as the equivalent value in a different integer type.
A Unicode string value that is a collection of characters.
```

C Interoperability

Operator Declarations

Tools for Your Types

Basic Behaviors

Encoding, Decoding and Serialization

Initialization with Literals

Values and Collections

Numbers and Basic Values

Logical Values

```
struct Bool
A value type whose instances are either true or false.
```

Numerical Values

```
struct Int
A signed integer value type.
```

```
struct Double
A double-precision, floating-point value type.
```

```
struct Float
A single-precision, floating-point value type.
```

Optionals

```
enum Optional
A type that represents either a wrapped value or nil, the absence of a value.
```

Ranges

```
struct Range
A half-open interval over a comparable type, from a lower bound up to, but not including, an upper bound.
```

```
struct Closed Range
An interval over a comparable type, from a lower bound up to, and including, an upper bound.
```

Error

```
protocol Error
A type representing an error value that can be thrown.
```

Advanced Numerics

Numeric Protocols

Special-Use Numeric Types such as UInt, Float80...

Global Numeric Functions such as func min<T>(T, T), func abs<T>(T), ...

Strings and Text

Work with text using Unicode-safe strings.

Strings and Characters

```
struct String
A Unicode string value that is a collection of characters.
```

```
struct Character
A single extended grapheme cluster that approximates a user-perceived character.
```

Compile-Time Strings

```
struct StaticString
A string type designed to represent text that is known at compile time.
```

Encoding and Storage

```
enum Unicode
A namespace for Unicode utilities.
```

Collections

Arrays & Dictionaries

```
struct Array
An ordered, random-access collection.
```

```
struct Dictionary
A collection whose elements are key-value pairs.
```

Sets

```
struct Set
An unordered collection of unique elements.
```

```
protocol OptionSet
A type that presents a mathematical set interface to a bit set.
```

Ranges

Create a collection of all the values in a range by using the half-open (.. $<$) and closed (... $>$) range operators.

```
func ..<Bound>(Bound, Bound)
Returns a countable half-open range that contains its lower bound but not its upper bound.
```

```
struct CountableRange
A half-open range that forms a collection of consecutive values.
```

```
func ...<Bound>(Bound, Bound)
Returns a countable closed range that
```

Special-Use Collections

These collections can store zero, one, or many of the same element.

```
func repeatElement<T>(T, count: Int)
Creates a collection containing the specified number of the given element.
```

```
struct CollectionOfOne
A collection containing a single element of type Element.
```

```
struct EmptyCollection
A collection whose element type is Element but that is always empty.
```

```
struct DictionaryLiteral
A lightweight collection of key-value pairs.
```

Joint Iteration

```
func zip<Sequence1, Sequence2>(Sequence1, Sequence2)
Creates a sequence of pairs built out of two underlying sequences.
```

Strides

Create a stride that steps over values between two boundaries using the stride(from:to:by:) and stride(from:through:by:) functions.

```
func stride<T>(from: T, to: T, by: T.Stride)
Returns the sequence of values (self, self + stride, self + 2 * stride, ... last) where last is the last value in the progression that is less than end.
```

```
func stride<T>(from: T, through: T, by: T.Stride)
Returns the sequence of values (self, self + stride, self + 2 * stride, ... last) where last is the last value in the progression less than or equal to end.
```

Sequences

Advanced Topics

Sequence and Collection Protocols

Supporting Types

Managed Buffers