Blog Download Getting Started Documentation

Documentation

The Swift Programming Language (6)

Topics

Welcome to Swift

About Swift

Understand the high-level goals of the language.

Version Compatibility

Learn what functionality is available in older language modes.

A Swift Tour

Explore the features and syntax of Swift.

Language Guide

The Basics

Work with common kinds of data and write basic syntax.

Basic Operators

Perform operations like assignment, arithmetic, and comparison.

Strings and Characters

Store and manipulate text.

Collection Types

Organize data using arrays, sets, and dictionaries.

Control Flow

Structure code with branches, loops, and early exits.

Functions

Define and call functions, label their arguments, and use their return values.

Closures

Group code that executes together, without creating a named function.

Enumerations

Model custom types that define a list of possible values.

Structures and Classes

Model custom types that encapsulate data.

Properties

Access stored and computed values that are part of an instance or type.

Methods

Define and call functions that are part of an instance or type.

Subscripts

Access the elements of a collection.

Inheritance

Subclass to add or override functionality.

Initialization

Set the initial values for a type’s stored properties and perform one-time setup.

Deinitialization

Release resources that require custom cleanup.

Optional Chaining

Access members of an optional value without unwrapping.

Error Handling

Respond to and recover from errors.

Concurrency

Perform asynchronous operations.

Macros

Use macros to generate code at compile time.

Type Casting

Determine a value’s runtime type and give it more specific type information.

Nested Types

Define types inside the scope of another type.

Extensions

Add functionality to an existing type.

Protocols

Define requirements that conforming types must implement.

Generics

Write code that works for multiple types and specify requirements for those types.

Opaque and Boxed Protocol Types

Hide implementation details about a value’s type.

Automatic Reference Counting

Model the lifetime of objects and their relationships.

Memory Safety

Structure your code to avoid conflicts when accessing memory.

Access Control

Manage the visibility of code by declaration, file, and module.

Advanced Operators

Define custom operators, perform bitwise operations, and use builder syntax.

Language Reference

About the Language Reference

Read the notation that the formal grammar uses.

Lexical Structure

Use the lowest-level components of the syntax.

Types

Use built-in named and compound types.

Expressions

Access, modify, and assign values.

Statements

Group expressions and control the flow of execution.

Declarations

Introduce types, operators, variables, and other names and constructs.

Attributes

Add information to declarations and types.

Patterns

Match and destructure values.

Generic Parameters and Arguments

Generalize declarations to abstract away concrete types.

Summary of the Grammar

Read the whole formal grammar.

Revision History

Document Revision History

Review the recent changes to this book.

Copyright © 2014–2023 Apple Inc. and the Swift project authors. All rights reserved.

This document is made available under a Creative Commons Attribution 4.0 International (CC BY 4.0) License.

Swift and the Swift logo are trademarks of Apple Inc.

Privacy Policy | Cookies

Light Dark Auto