

Язык программирования Swift

Прочитайте разделы, описанные ниже, в документации по языку программирования Swift.

Целью задания на чтение, является улучшение ваших знаний языка. Важные понятия, которые необходимо изучить на данный момент выделены тремя цветами:

Красные разделы **ОЧЕНЬ ВАЖНЫ** и могут быть более сложными для понимания. Читать это тщательно.

Желтые разделы важны, но, вероятно, их будет не так сложно понять.

Затененные разделы не требуются для чтения (на этой неделе). Они могут быть в ближайшие недели.

Не замазывайте чтение текста ПРИМЕЧАНИЕ (внутри серых рамок) - многие из этих вещей довольно важно. Однако, если ПРИМЕЧАНИЕ ссылается на Objective-C или мостовое соединение, вы можете его игнорировать.

Если в тексте есть ссылка на другой раздел, вам не нужно переходить по этой ссылке, если то, на что он ссылается, также является частью задания по чтению на этой неделе.

Связанные темы не более или менее важны, чем любая другая тема.

В области «Руководство по языку» прочитайте следующие разделы в следующих главах:

Русско-язычная документация

The Basics

Constants and Variables

Comments

Semicolons

Integers

Floating-Point Numbers

Type Safety and Type Inference

Numeric Literals

Numeric Type Conversion

Type Aliases

Booleans

Tuples

Optionals

Error Handling

Assertions and Preconditions

Basic Operators

Terminology

Assignment Operator

Arithmetic Operators

Compound Assignment Operators

Comparison Operators

Ternary Conditional Operator

Nil-Coalescing Operator

Range Operators

Logical Operators

Strings and Characters

String Literals

Initializing an Empty String

String Mutability

Strings Are Value Types

Working with Characters

Concatenating Strings and Characters

String Interpolation

Unicode

Counting Characters

Accessing and Modifying a String

Substrings

Comparing Strings

Unicode Representations of Strings

Collection Types

Mutability of Collections

Arrays

Sets

Performing Set Operations

Dictionaries

Control Flow

For-In Loops

While Loops

Conditional Statements

If

Switch

No Implicit Fallthrough

Interval Matching

Tuples

Value Bindings

Where

Compound Cases

Control Transfer Statements

Continue

Break

Fallthrough

Labeled Statements

Early Exit

Checking API Availability

Functions

Defining and Calling Functions

Function Parameters and Return Values

Functions Without Parameters

Functions With Multiple Parameters

Functions Without Return Values

Functions With Multiple Return Values

Function Argument Labels and Parameter Names

Specifying Argument Labels

Omitting Argument Labels

Default Parameter Values

Variadic Parameters

In-Out Parameters

Function Types

Nested Functions

Closures

Closure Expressions

Trailing Closures

Capturing Values

Closures Are Reference Types

Escaping Closures

Autoclosures

Enumerations

Enumeration Syntax

Matching Enumeration Values with a Switch Statement

Associated Values

Raw Values

Recursive Enumerations

Classes and Structures

Comparing Classes and Structures

Structures and Enumerations Are Value Types (ignore enumerations)

Classes Are Reference Types

Choosing Between Classes and Structures

Assignment and Copy Behavior for Strings, Arrays and Dictionaries

Properties

Stored Properties

Computed Properties

Property Observers

Global and Local Variables

Type Properties

Methods

Instance Methods

The self Property

Modifying Value Types from Within Instance Methods

Assigning to self Within a Mutating Method

Type Methods

Subscripts

Inheritance

Defining a Base Class

Subclassing

Overriding

Accessing Superclass Methods, Properties, and Subscripts

Overriding Methods

Overriding Properties

Preventing Overrides

Initialization

Setting Initial Values for Stored Properties

Customizing Initialization

Default Initializers

Class Inheritance and Initialization

Failable Initializers

Required Initializers

Deinitialization

Optional Chaining

Error Handling

Type Casting

Nested Types

Extensions

Protocols

Automatic Reference Counting

Memory Safety

Access Control

Advanced Operators