



# RESOURCES FOR LEARNING iOS

approved by NIX iOS experts

You can get started creating mobile apps for iOS by learning the popular Swift programming language. Apple developers have been struggling for years to create a simple, safe, and convenient language. Swift has absorbed the best ideas of modern languages with the wisdom of Apple's engineering culture. The compiler is tuned for performance and the language is tuned for development, with no middle ground on either side. Swift is beginner-friendly for programming newbies. It is the first industrial-quality programming language to be as easy to understand and fun to use as a scripting language. Writing code in a sandbox allows you to tinker with Swift code and instantly see the result, without having to compile and run the program. Well, let's discover the world of Apple technologies!

## BASICS

1. You should start learning the language with a close look at the official [documentation from Apple](#)

Pay particular attention to the following:

- Memory management and operation;
- Classes and structures (specifics of creation and use);
- Optionals, their use, and objectives;
- Types of collections, what is the difference and features of use;
- Functions and closures.

*Other specifics of Swift are no less important for full-fledged iOS programming and understanding them will be a great advantage for the training.*

1. Be sure to secure the acquired knowledge through practice and experiments in [Swift Playgrounds](#)

## BEST PRACTICES AND BASICS

1. Novice developers should be familiar with all the basic principles of [OOP](#).
2. Understanding as many [design patterns](#) as possible will be a huge plus.
3. Skills in [SOLID](#) will also be a plus.
4. In his work, every developer may face multithreading and competitive execution. Therefore, knowing the basic concepts from this area, which are described in the article [“Multithreading and Concurrency Fundamentals”](#), will be an indicator of technical literacy.

*Think beyond the references provided and look for different sources to understand the approaches from different angles and on different examples. Search or invent tasks in any subject area and try to solve them using the principles you have learned.*

## UI COMPONENTS

It is crucial to figure out the functioning of any UI component, such as UITableView. There are many tutorials about them, such as this one: [“Adding a UITableView programmatically”](#)

Moreover, be sure to understand:

- How to draw table cells;
- How they are reused;
- What mandatory methods should be implemented;
- What are the convenience and benefits of such a mechanism;
- What is the lifecycle of the program, and what state it passes from;
- What is the lifecycle of UIViewController;
- What is the difference between UIWindow and UIView.

*You may find it helpful to try everything with your small test application to be able to show the code and tell why certain solutions were used.*



**WE ARE WAITING FOR YOU!**



<https://nixstech.com/>