

Functional Programming in Swift

Course Summary

Description

This course will encourage you to look at functions and types differently. You can opt for the one day or two day course.

Day one focuses on higher-order functions and some of the principles of functional programming. We will create functions that return and accept other functions and then explore examples of higher-order functions in the Swift Standard Library's Arrays including `filter()`, `sorted()`, `reduce()`, `map()`, `compactMap()`, `flatMap()`, and `zip()`.

The second day focuses on `map()` and `flatMap()` as design patterns. You'll see examples of functors and monads beginning with Arrays, Optionals, and Result, and moving on to Writer, Reader, and State.

We assume that you are an experienced programmer who has spent some time looking at Swift but you are relatively new to Functional Programming.

Topics

- Immutability and Functions
- Higher-Order Functions in Arrays
- Map (Functors)
- FlatMap (Monads)

Audience

This course is for experienced Swift developers who want to explore Functional Programming.

Prerequisites

We assume that you are an experienced programmer who has spent some time looking at Swift but you are relatively new to Functional Programming.

Duration

One day

Functional Programming in Swift

Course Outline

I. Immutability and Functions

- A. Issues with mutability
- B. Non-mutating methods
- C. Generic Model
- D. Lines
- E. Making Functions
- F. Closures
- G. Higher-Order Functions
- H. Call as Function

II. Higher-Order Functions in Arrays

- A. Filter
- B. Sorted
- C. Map
- D. CompactMap
- E. FlatMap
- F. Reduce
- G. Zip

III. Map (Functors)

- A. Arrays
- B. One at a Time
- C. Zero or One
- D. Optionals
- E. Result
- F. Writer
- G. Reader
- H. State

IV. FlatMap (Monads)

- A. Arrays
- B. Writer
- C. Result
- D. Optionals
- E. Just
- F. Reader
- G. State