Swift & SwiftUI

Computer programing II
Applied Mathematics, National Dong Hwa University

Jiann-Ming Wu, Feb 2023

This course introduces fundamental swift programming and SwiftUI App design. Programming here is object-oriented and extensively integrated with user interface of iOS components and touch screen views.

It will first give introduction to new features of the swift programming language as well as swift basics, true or false, if statements, optional , arrays, loop de loop, string, dictionary, functions, switch, tuples, enums, classes, closures, extensions, and properties.

SwiftUI programing on Xcode will be also introduced for iPhone App design. Basic and advanced artificial intelligence App design examples will be also demonstrated.

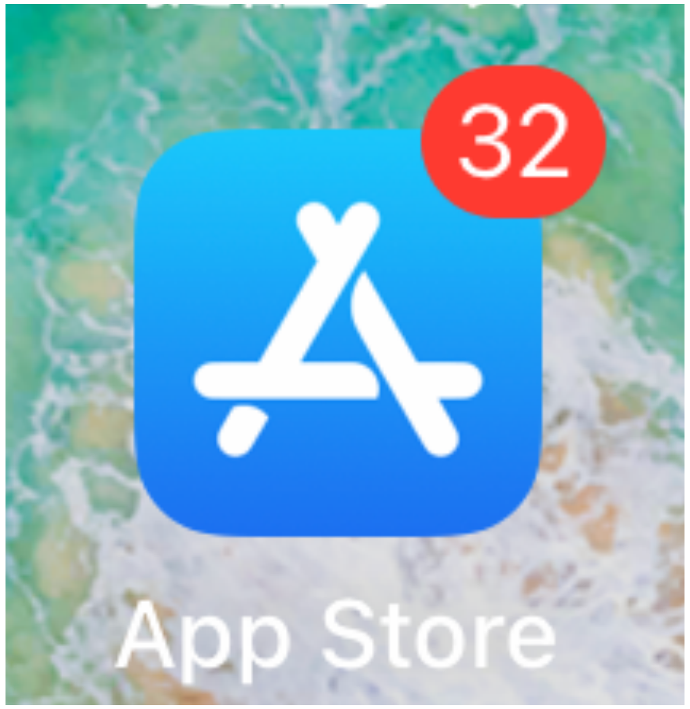
Course Teaching Website

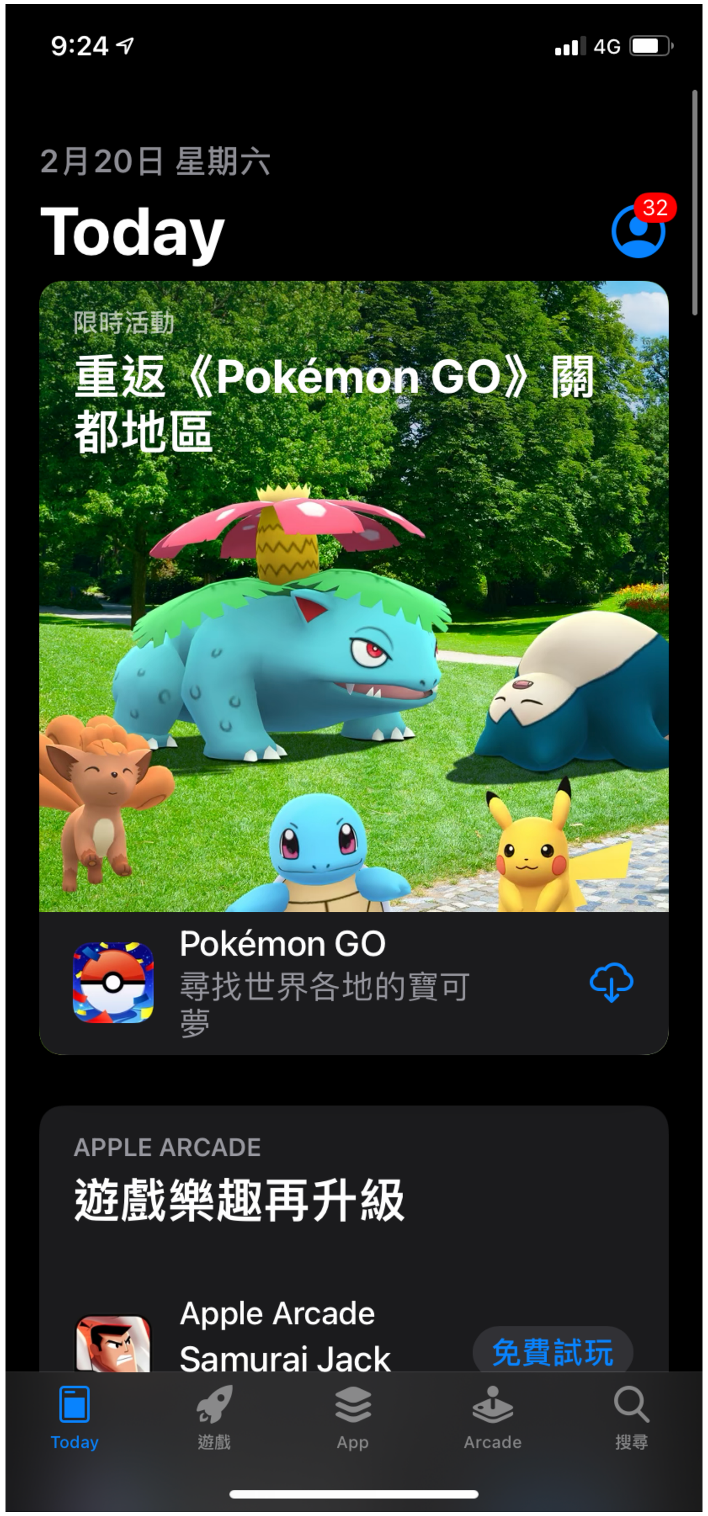
http://gpfai.net

Apple App Store

wikipedia App_Store_(iOS/iPadOS)

- Let everybody share Apps
- Get Apps
- Applications on iOS devices
-





2月20日 星期六

Today

限時活動

重返《Pokémon GO》關都地區

Pokémon GO

尋找世界各地的寶可夢

APPLE ARCADE

遊戲樂趣再升級

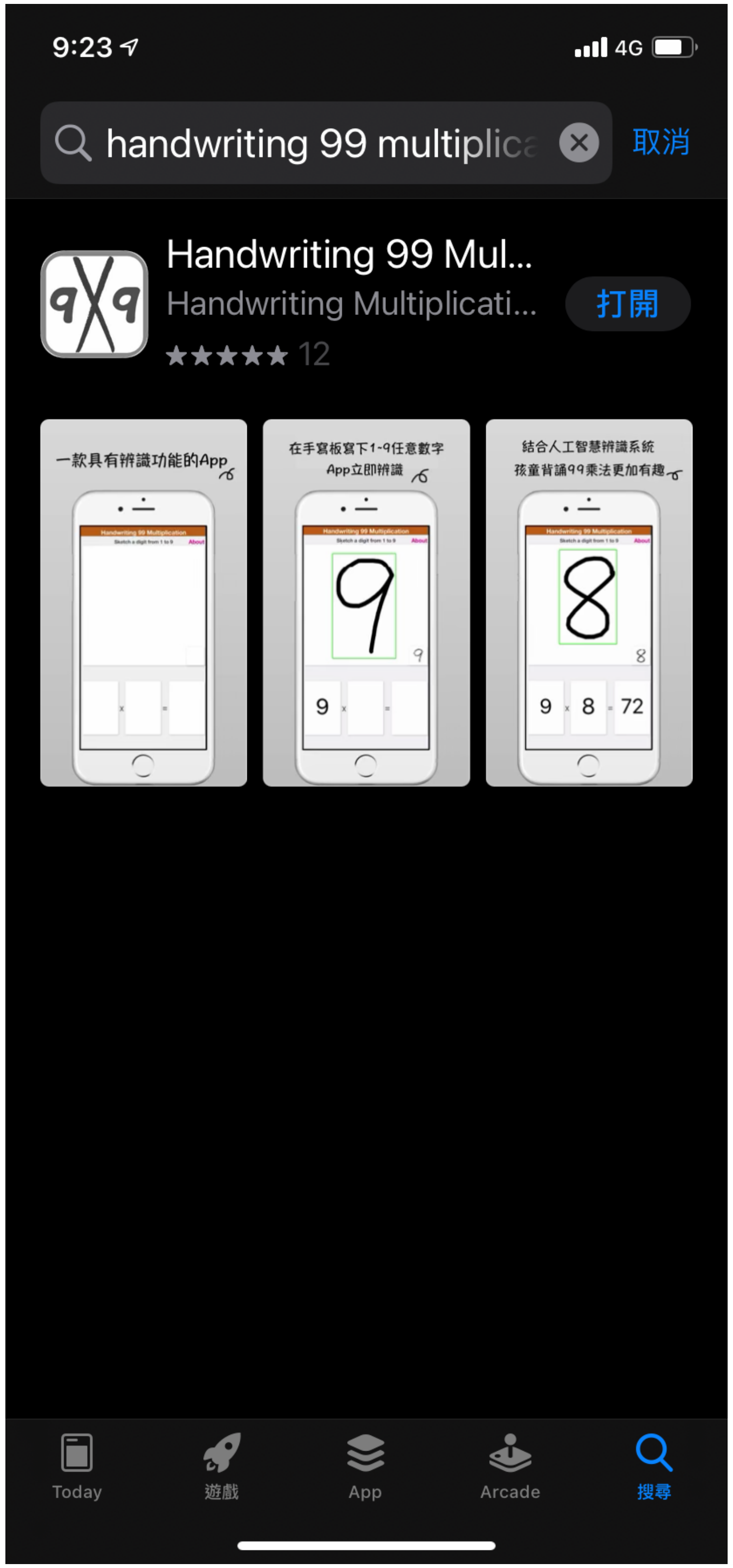
Apple Arcade

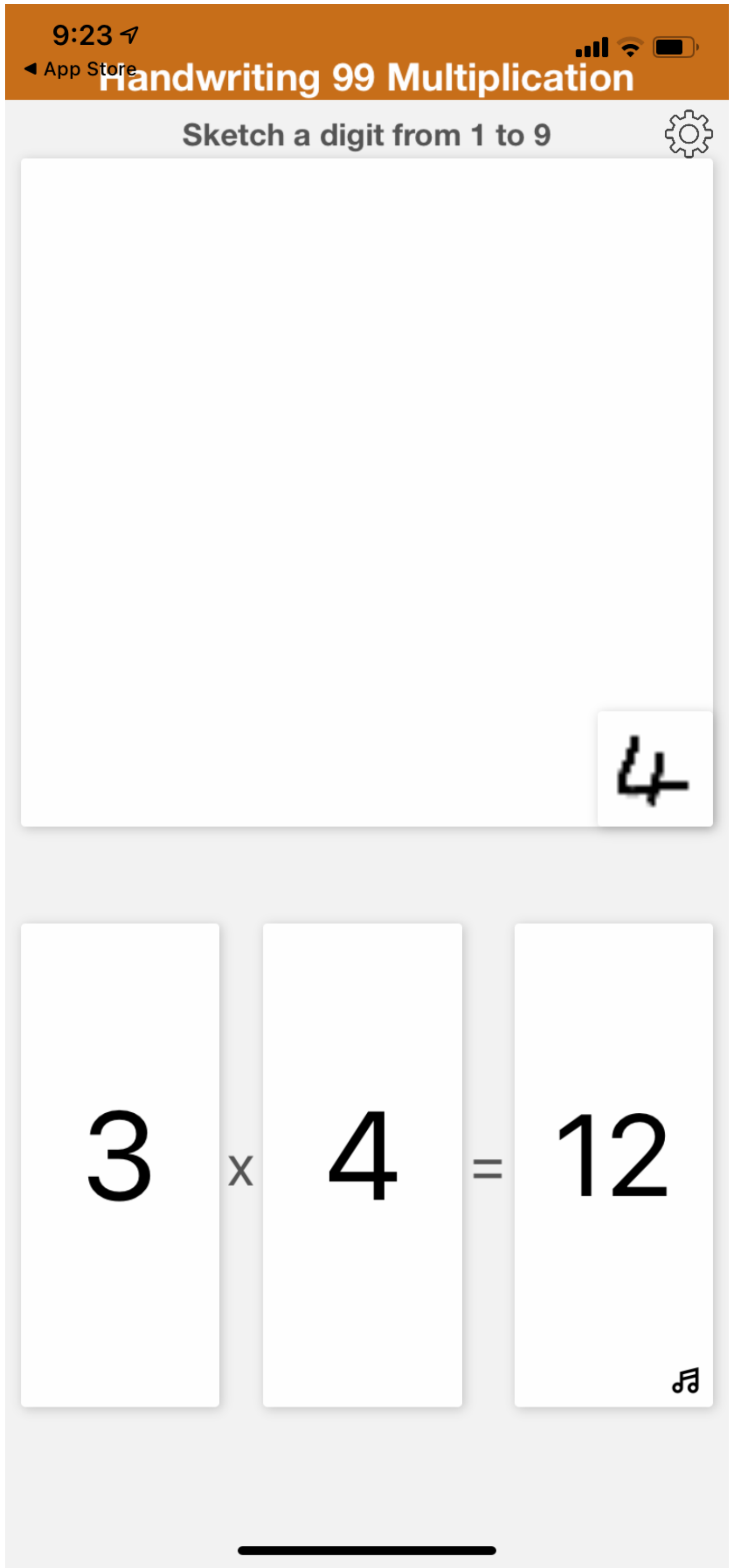
Samurai Jack

AI App

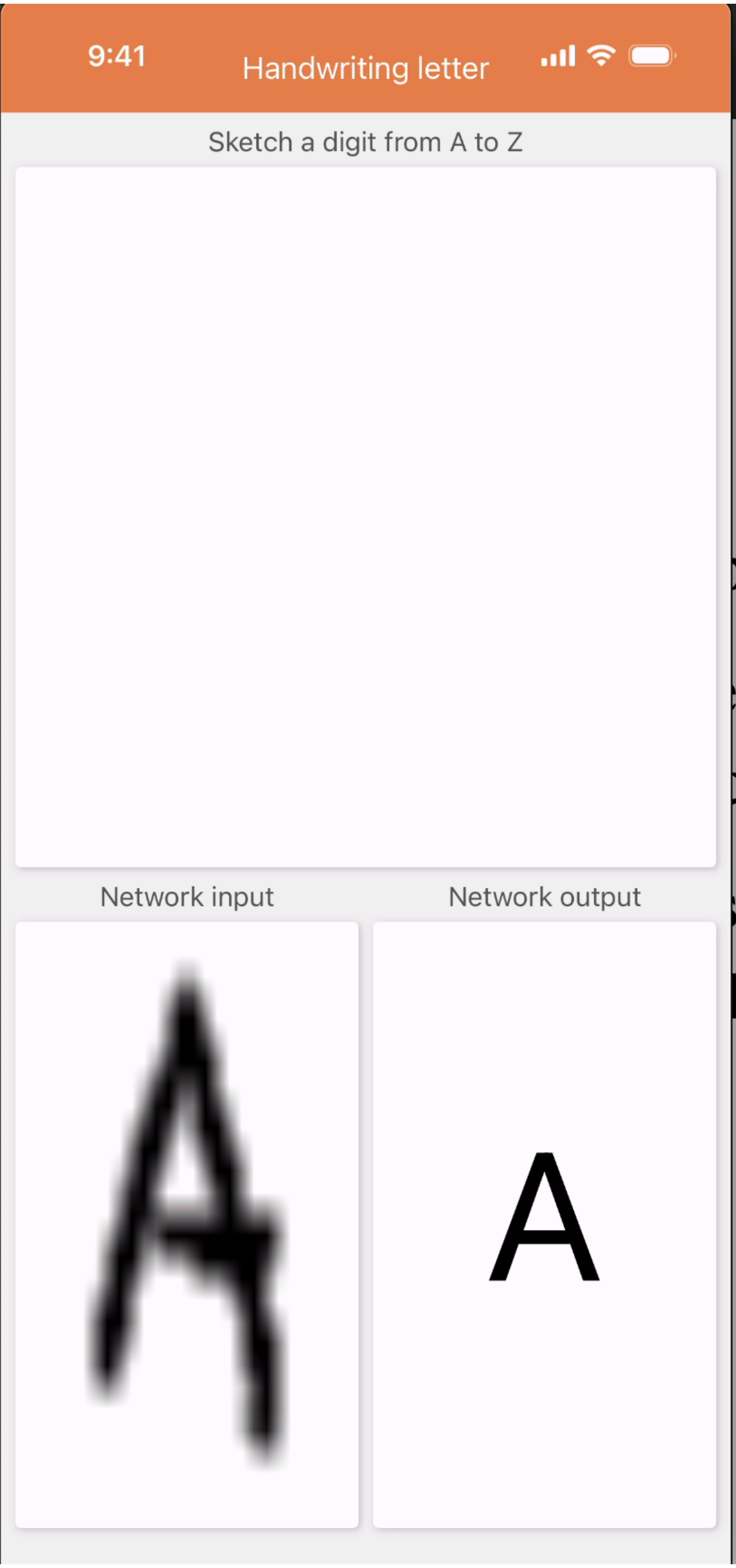
Handwriting 99 multiplication

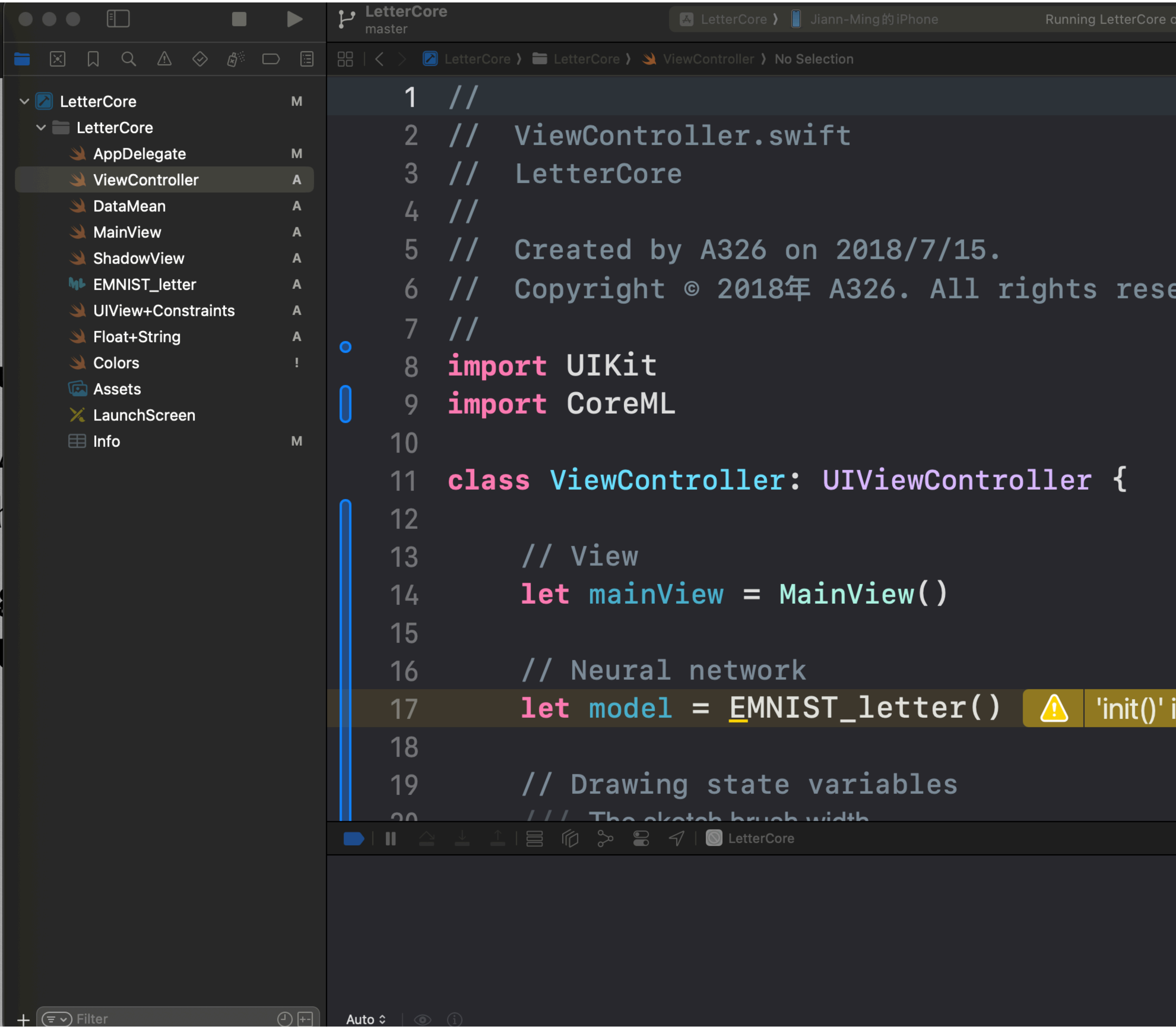
- An App published
- Integrate AI CNN models with touch screen of iPhone
- Author is graduated from AM NDHU
- 2018





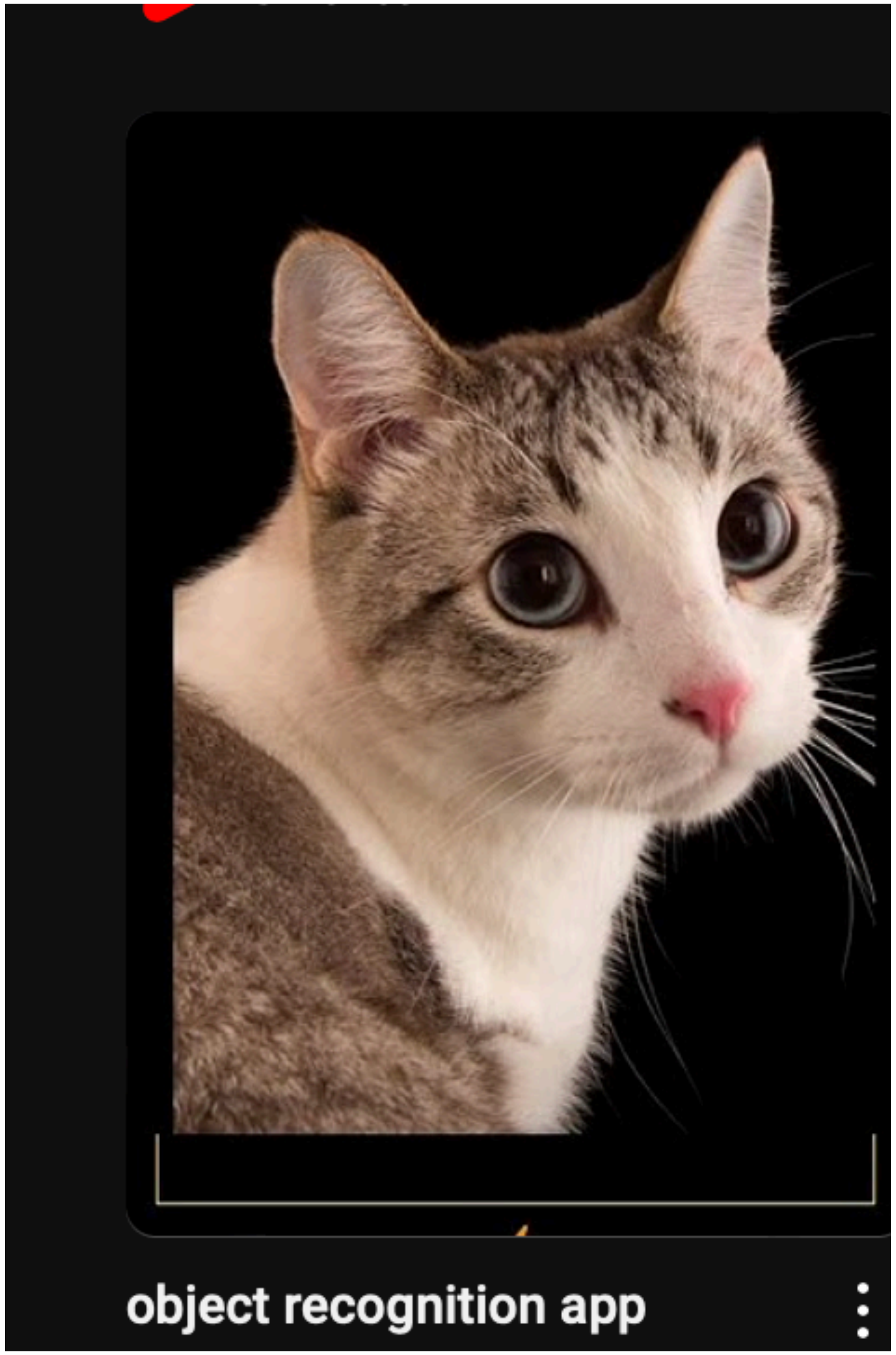
Install to iPhone and execute on iPhone





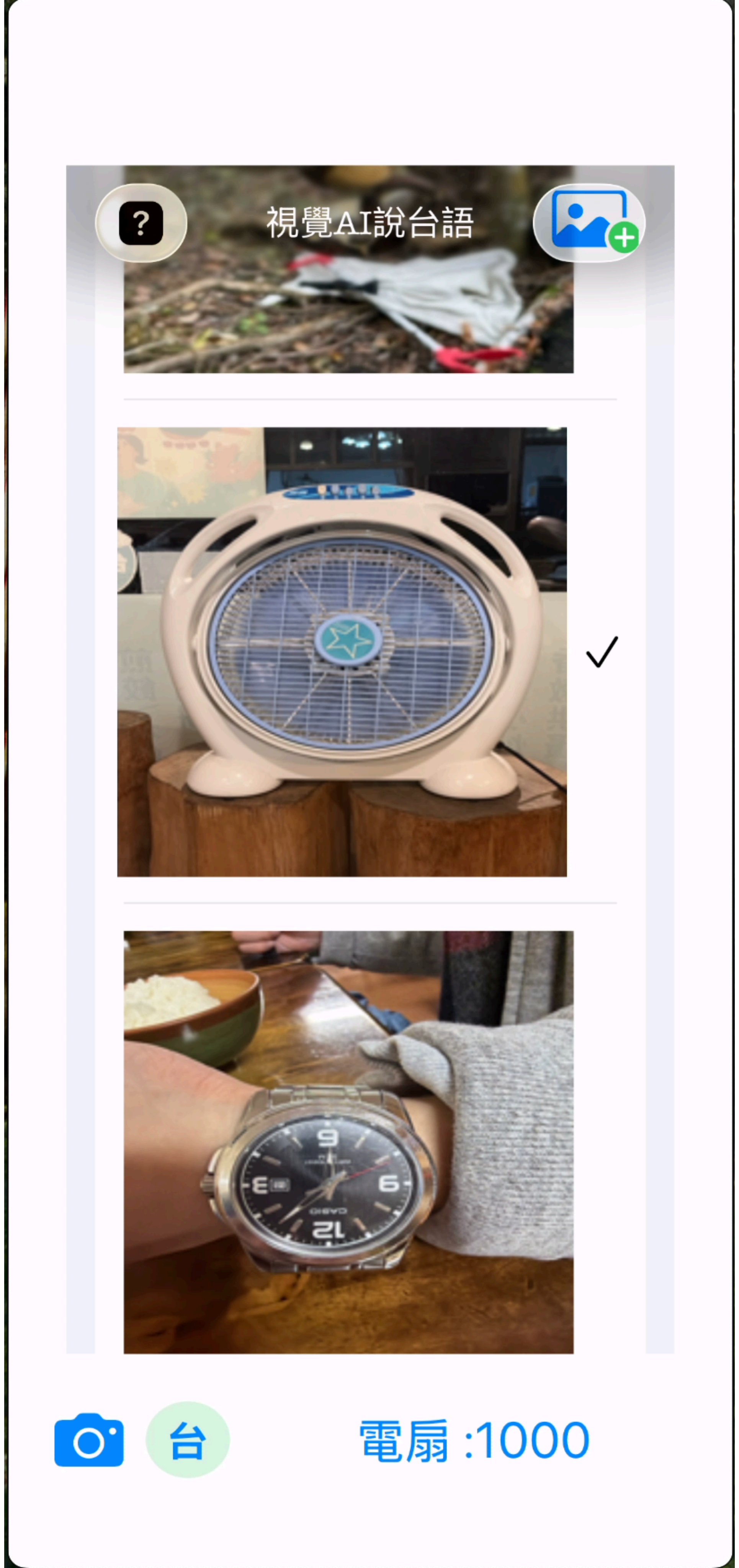
```
//
//  ViewController.swift
//  LetterCore
//
//  Created by A326 on 2018/7/15.
//  Copyright © 2018年 A326. All rights rese
//
import UIKit
import CoreML

class ViewController: UIViewController {

    // View
    let mainView = MainView()

    // Neural network
    let model = EMNIST_letter()   ⚠ 'init()' i

    // Drawing state variables
    /// The sketch brush width
```

AI Object Recognition App

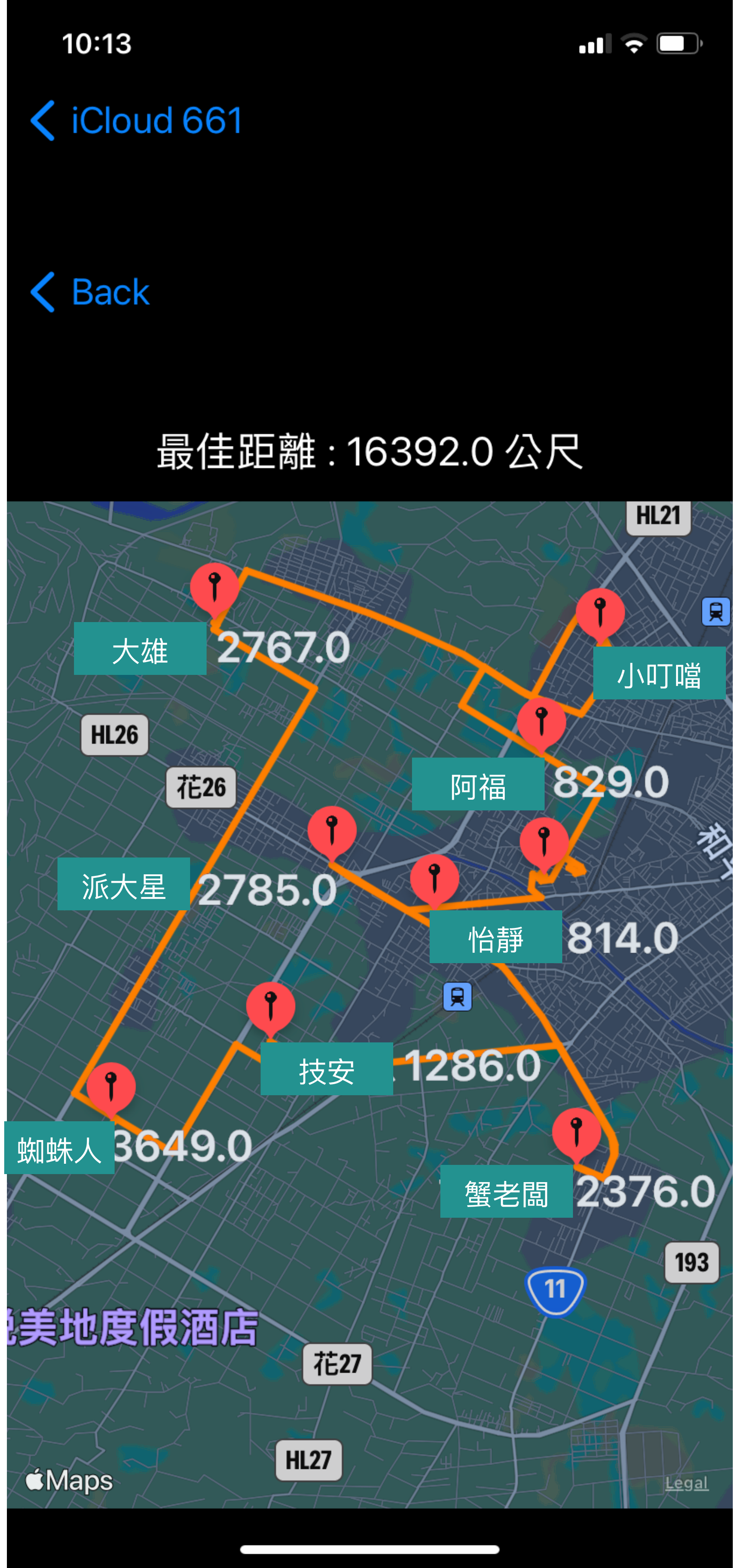


object recognition app

Execute an App: 視覺AI說台語



Shortest Path Optimization App



10:13

iCloud 661

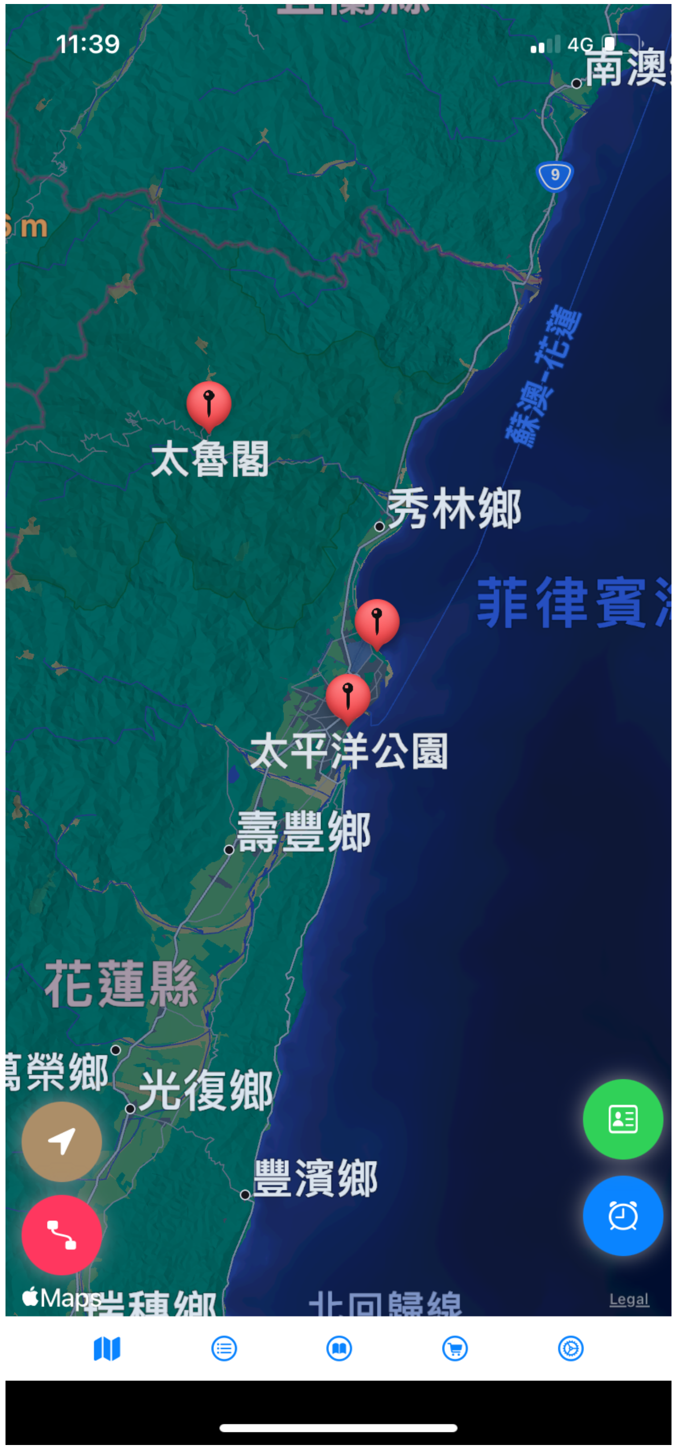
Back

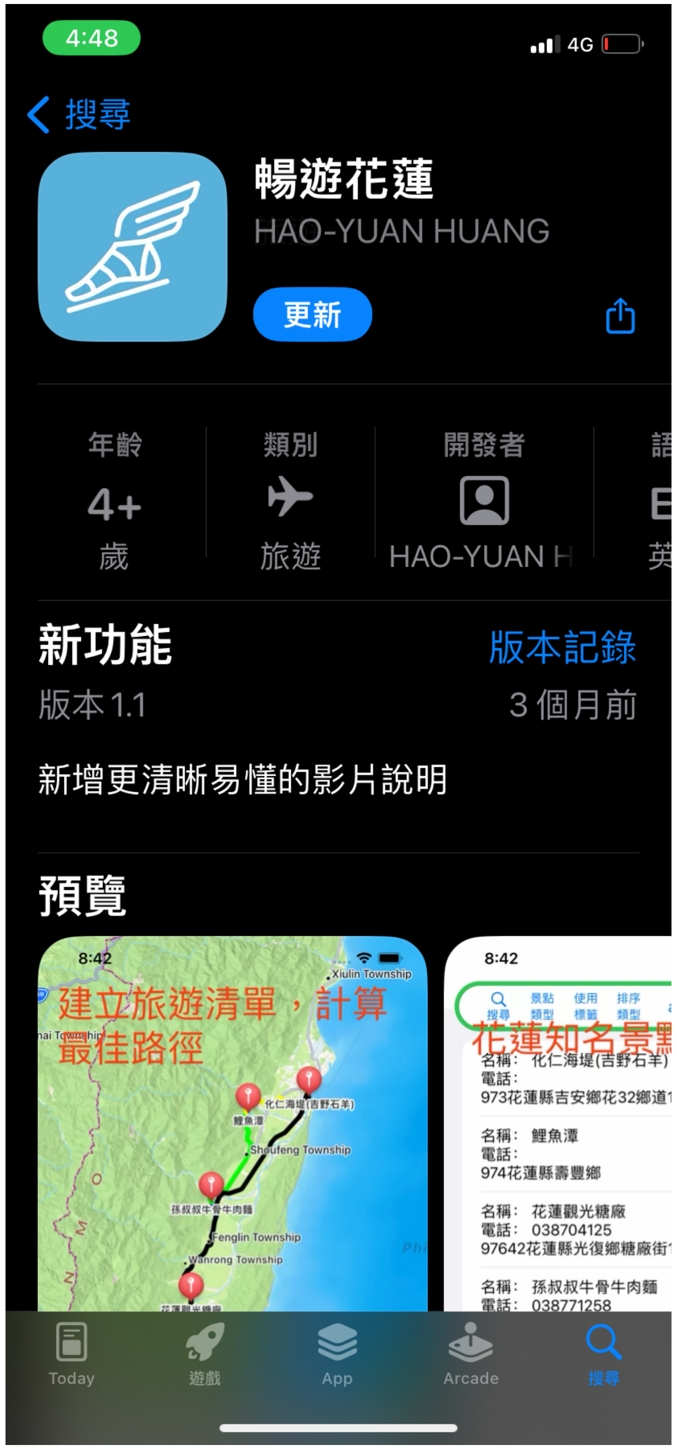
最佳距離：16392.0 公尺

AI App

暢遊花蓮

- An App published
- Integrate AI CNN models with touch screen of iPhone
- Author is from AM NDHU
- 2023





4:48 4G

< 搜尋

暢遊花蓮

HAO-YUAN HUANG

更新

年齡	類別	開發者	語
4+	✈		E
歲	旅遊	HAO-YUAN H	英

新功能 版本記錄

版本 1.1 3 個月前

新增更清晰易懂的影片說明

預覽

8:42

Xiulin Township

建立旅遊清單，計算最佳路徑

化仁海堤(吉野石羊)

鯉魚潭

Shoufeng Township

孫叔叔牛骨牛肉麵

Fenglin Township

Wanrong Township

8:42

搜尋 景點類型 使用標籤 排序類型

花蓮知名景點

名稱： 化仁海堤(吉野石羊)
電話：
973花蓮縣吉安鄉花32鄉道1

名稱： 鯉魚潭
電話：
974花蓮縣壽豐鄉

名稱： 花蓮觀光糖廠
電話： 038704125
97642花蓮縣光復鄉糖廠街1

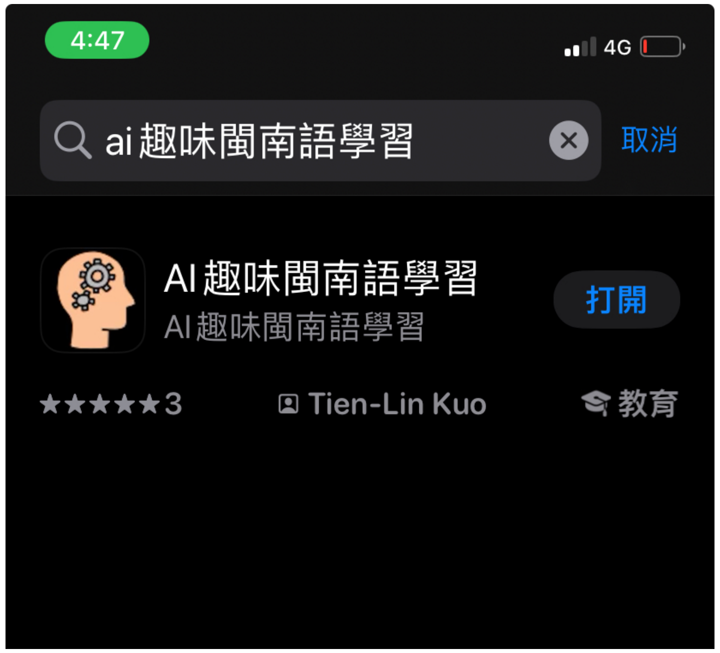
名稱： 孫叔叔牛骨牛肉麵
電話： 038771258

Today 遊戲 App Arcade 搜尋

AI App

AI趣味閩南語學習

- An App published
- Integrate AI CNN models with touch screen of iPhone
- Author is from AM NDHU
- 2023



4:47

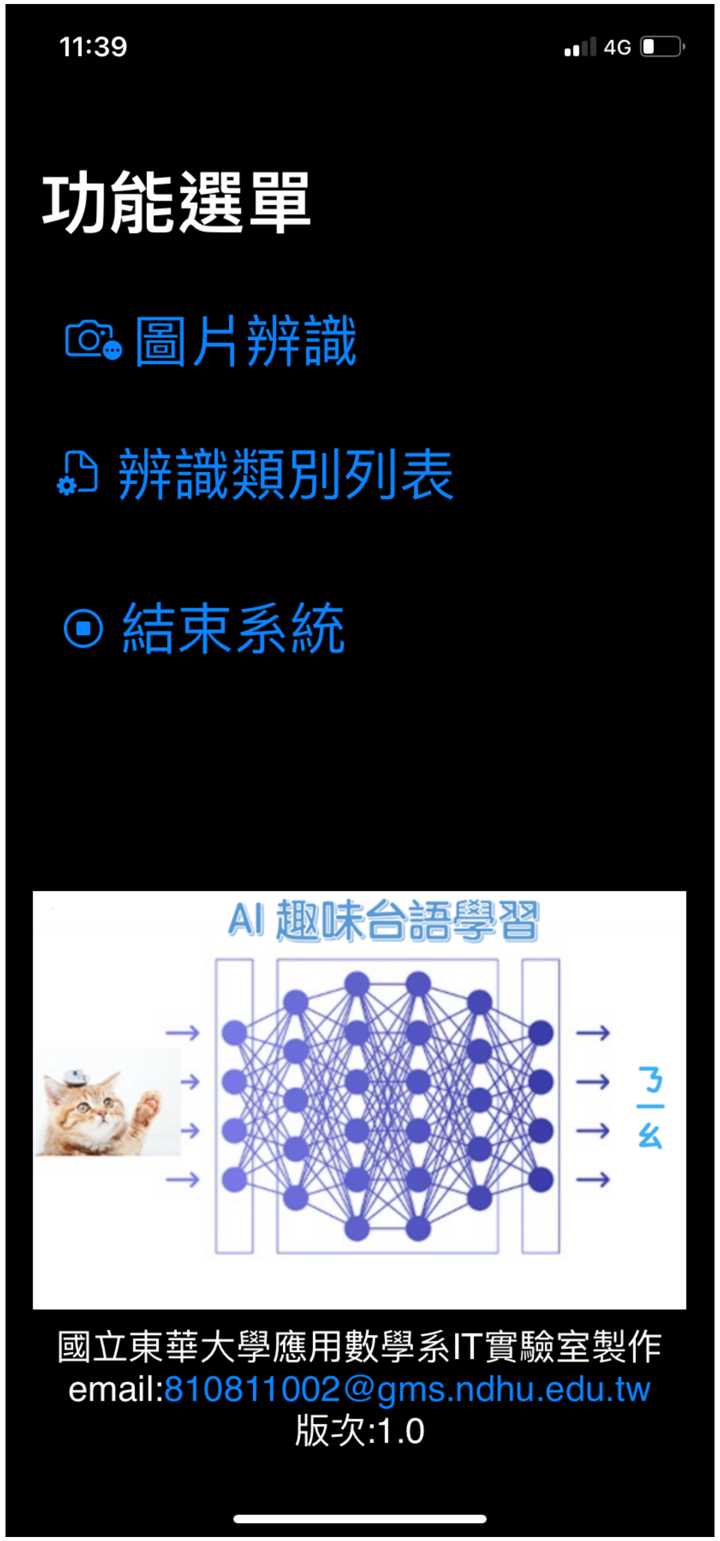
ai 趣味閩南語學習 取消

AI 趣味閩南語學習

AI 趣味閩南語學習

打開

★★★★★3 Tien-Lin Kuo 教育



11:39

功能選單

圖片辨識

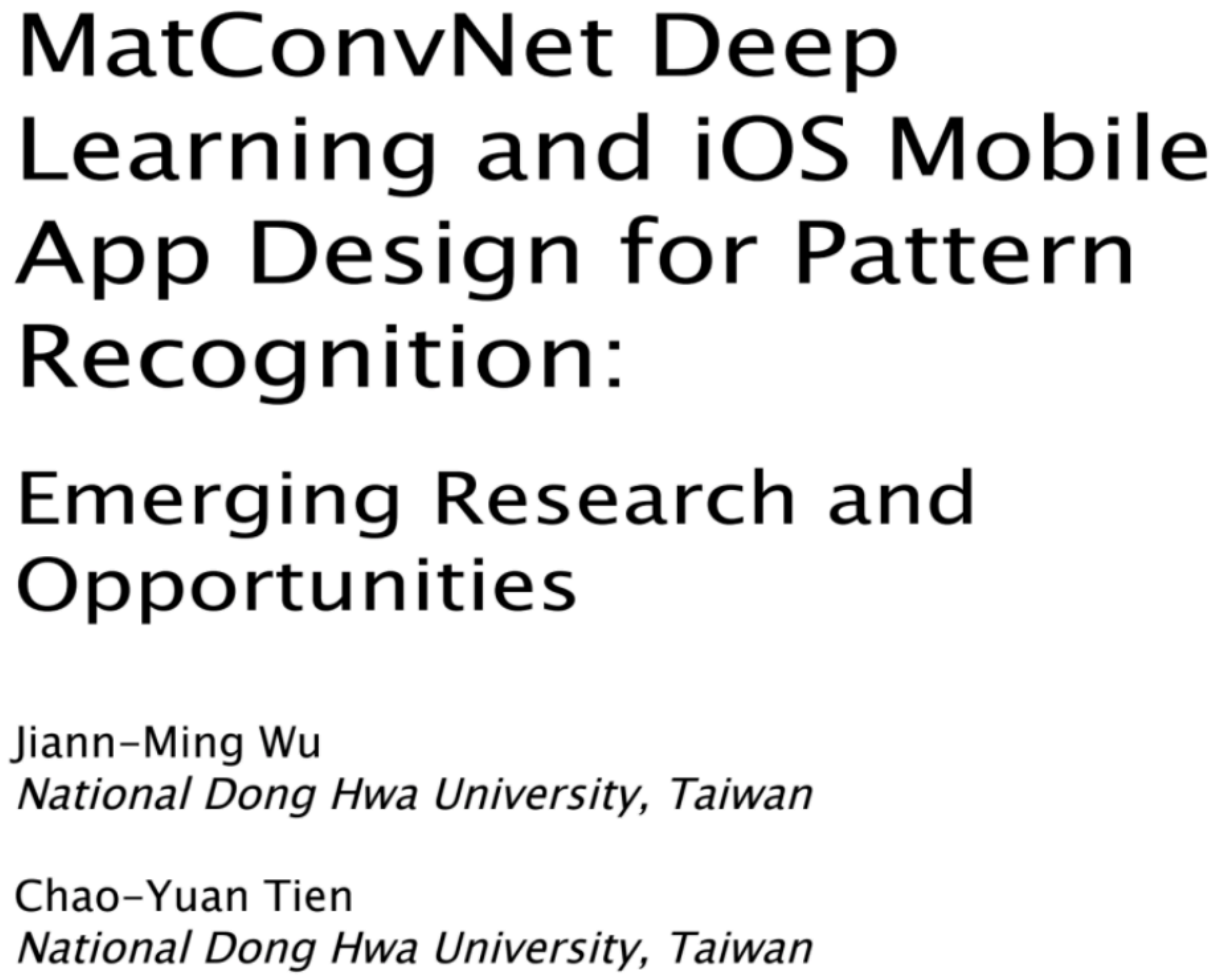
辨識類別列表

結束系統

國立東華大學應用數學系IT實驗室製作
email:810811002@gms.ndhu.edu.tw
版次:1.0

A Book

Deep learning has become a trending area of research due to its adaptive characteristics and high levels of applicability. In recent years, researchers have begun applying deep learning strategies to image analysis and pattern recognition for solving technical issues within image classification. As these technologies continue to advance, professionals have begun translating this intelligent programming language into mobile applications for devices. Programmers and web developers are in need of significant research on how to successfully develop pattern recognition applications using intelligent programming.

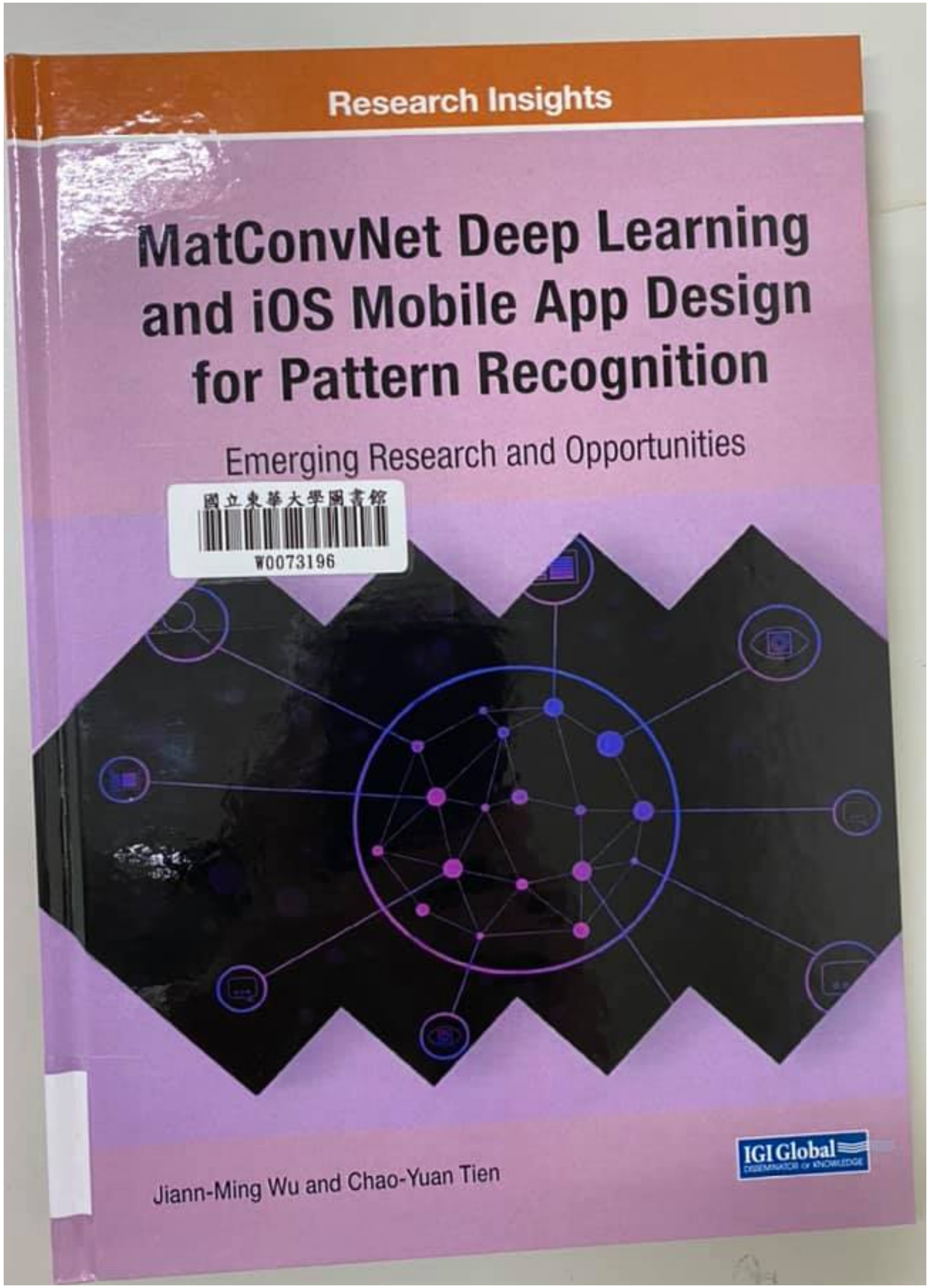


MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition:

Emerging Research and Opportunities

Jiann-Ming Wu
National Dong Hwa University, Taiwan

Chao-Yuan Tien
National Dong Hwa University, Taiwan



Research Insights

MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition

Emerging Research and Opportunities

Jiann-Ming Wu and Chao-Yuan Tien

IGI Global
DISSEMINATOR of KNOWLEDGE

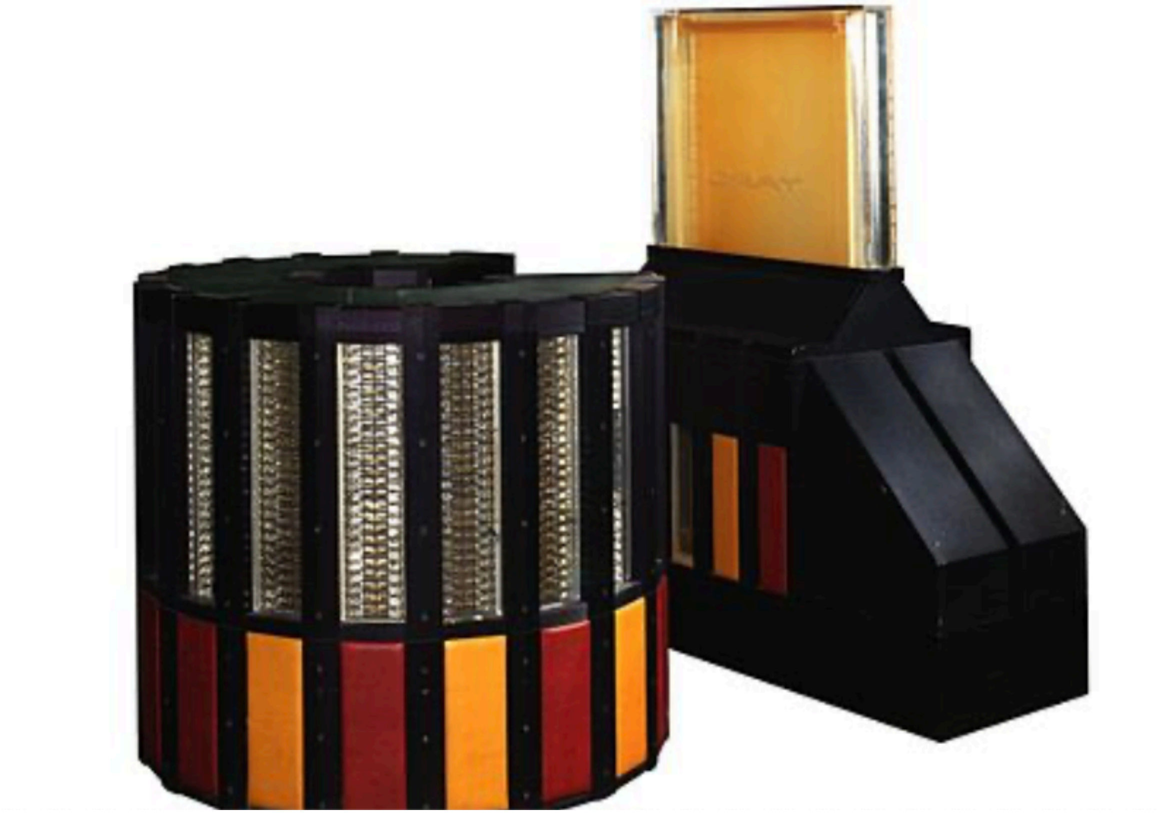
MatConvNet Deep Learning and iOS Mobile App Design for Pattern Recognition: Emerging Research and Opportunities is an essential reference source that presents a solution to developing intelligent pattern recognition Apps on iOS devices based on MatConvNet deep learning. Featuring research on topics such as medical image diagnosis, convolutional neural networks, and character classification, this book is ideally designed for programmers, developers, researchers, practitioners, engineers, academicians, students, scientists, and educators seeking coverage on the specific development of iOS mobile applications using pattern recognition strategies.

Cary 2

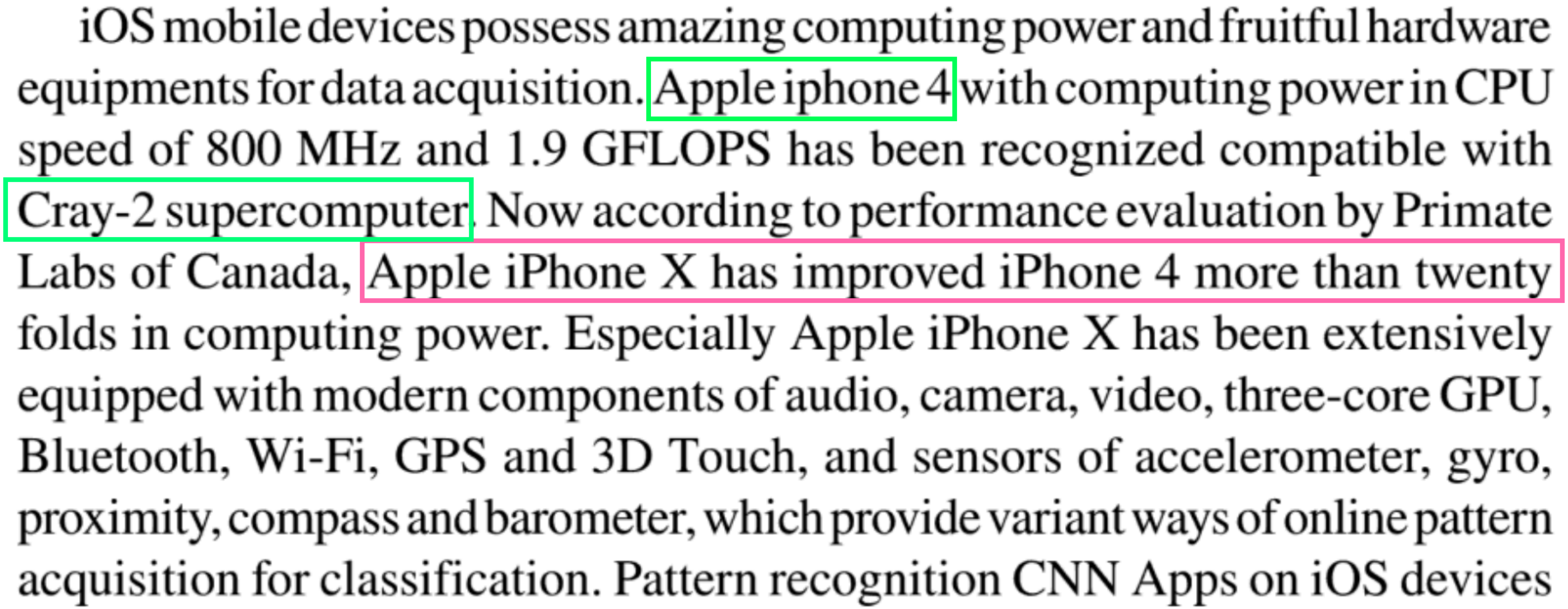
Supercomputer

The **Cray-2** is a supercomputer with four vector processors made by Cray Research starting in 1985. At 1.9 GFLOPS peak performance, it was the fastest machine in the world when it was released, replacing the Cray X-MP in that spot. It was, in turn, replaced in that spot by the Cray Y-MP in 1988.

Cray-2

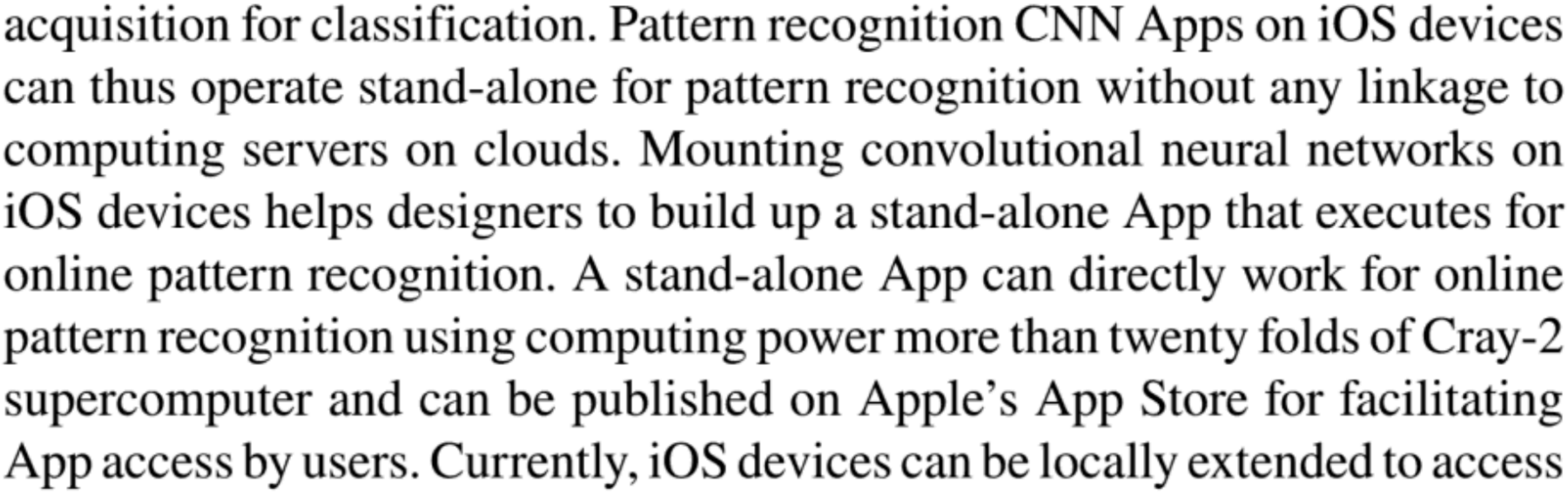


Iphone X and Cray II



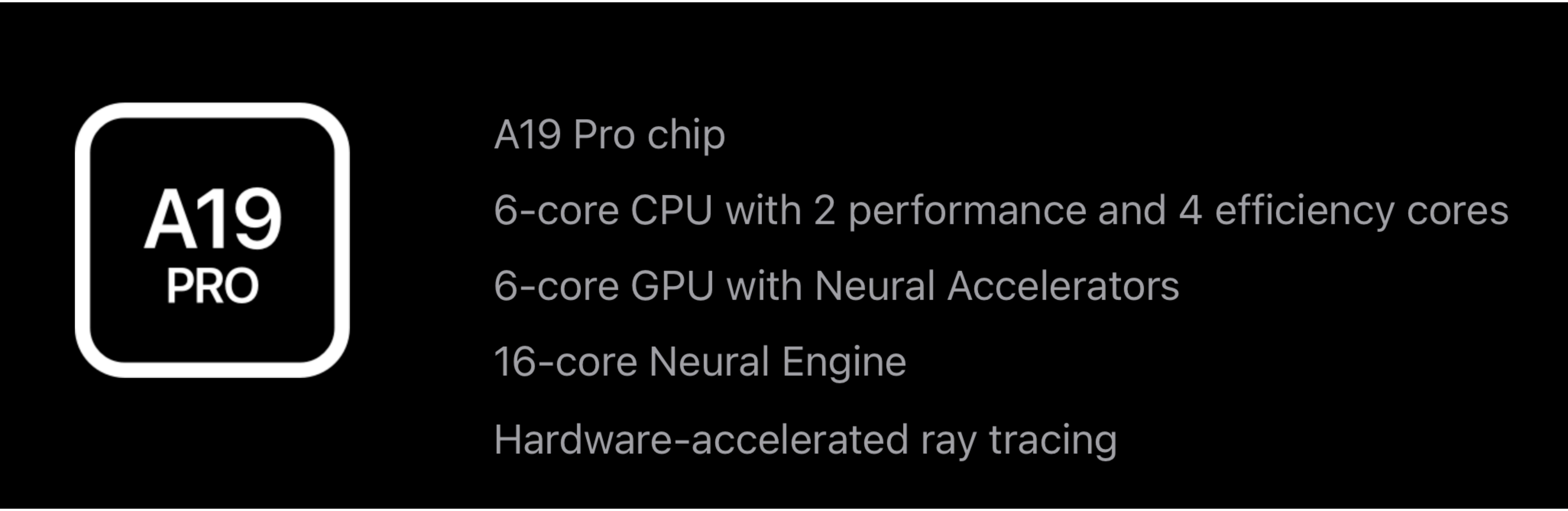
iOS mobile devices possess amazing computing power and fruitful hardware equipments for data acquisition. Apple iphone 4 with computing power in CPU speed of 800 MHz and 1.9 GFLOPS has been recognized compatible with Cray-2 supercomputer. Now according to performance evaluation by Primate Labs of Canada, Apple iPhone X has improved iPhone 4 more than twenty folds in computing power. Especially Apple iPhone X has been extensively equipped with modern components of audio, camera, video, three-core GPU, Bluetooth, Wi-Fi, GPS and 3D Touch, and sensors of accelerometer, gyro, proximity, compass and barometer, which provide variant ways of online pattern acquisition for classification. Pattern recognition CNN Apps on iOS devices

Standalone AI App on Iphones



acquisition for classification. Pattern recognition CNN Apps on iOS devices can thus operate stand-alone for pattern recognition without any linkage to computing servers on clouds. Mounting convolutional neural networks on iOS devices helps designers to build up a stand-alone App that executes for online pattern recognition. A stand-alone App can directly work for online pattern recognition using computing power more than twenty folds of Cray-2 supercomputer and can be published on Apple's App Store for facilitating App access by users. Currently, iOS devices can be locally extended to access

iPhone 17 pro max



A19 Pro chip

6-core CPU with 2 performance and 4 efficiency cores

6-core GPU with Neural Accelerators

16-core Neural Engine

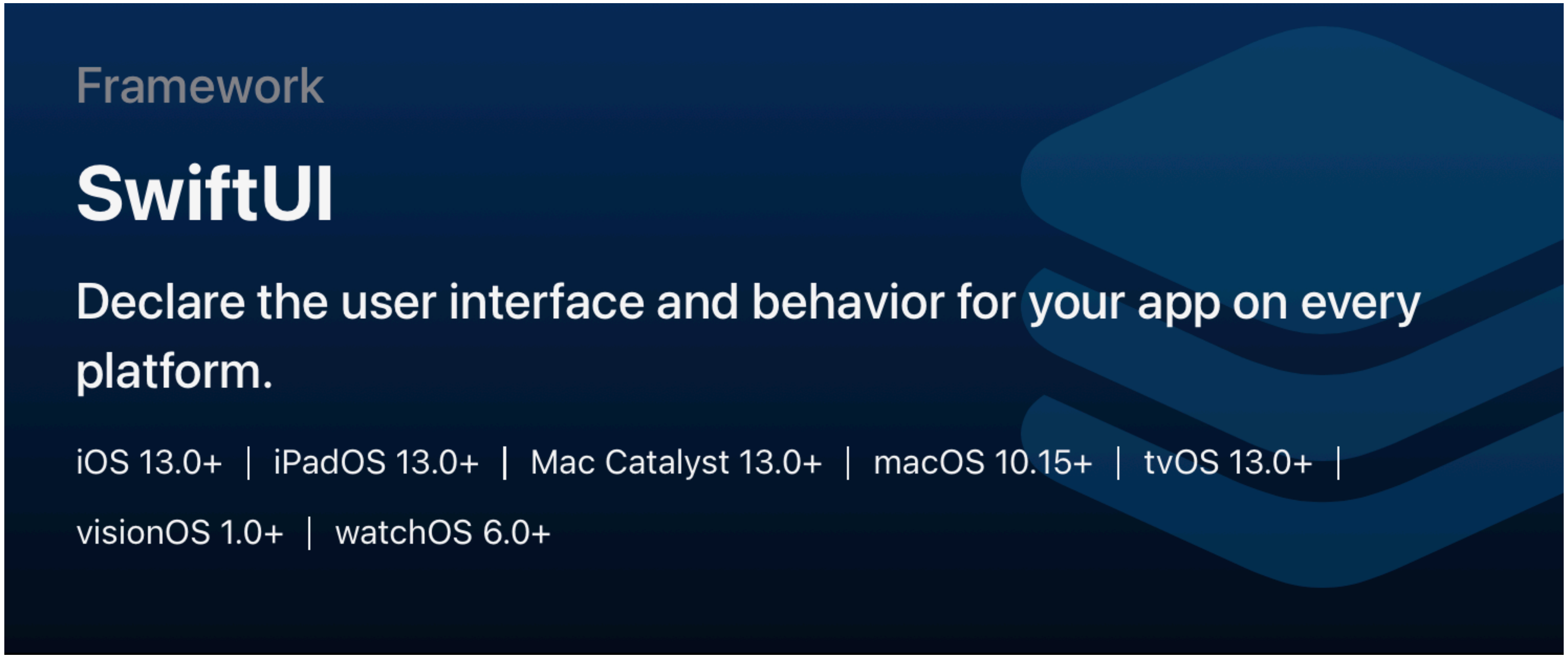
Hardware-accelerated ray tracing

SwiftUI

Easy to design Apps on iOS devices

SwiftUI

Better apps. Less code.

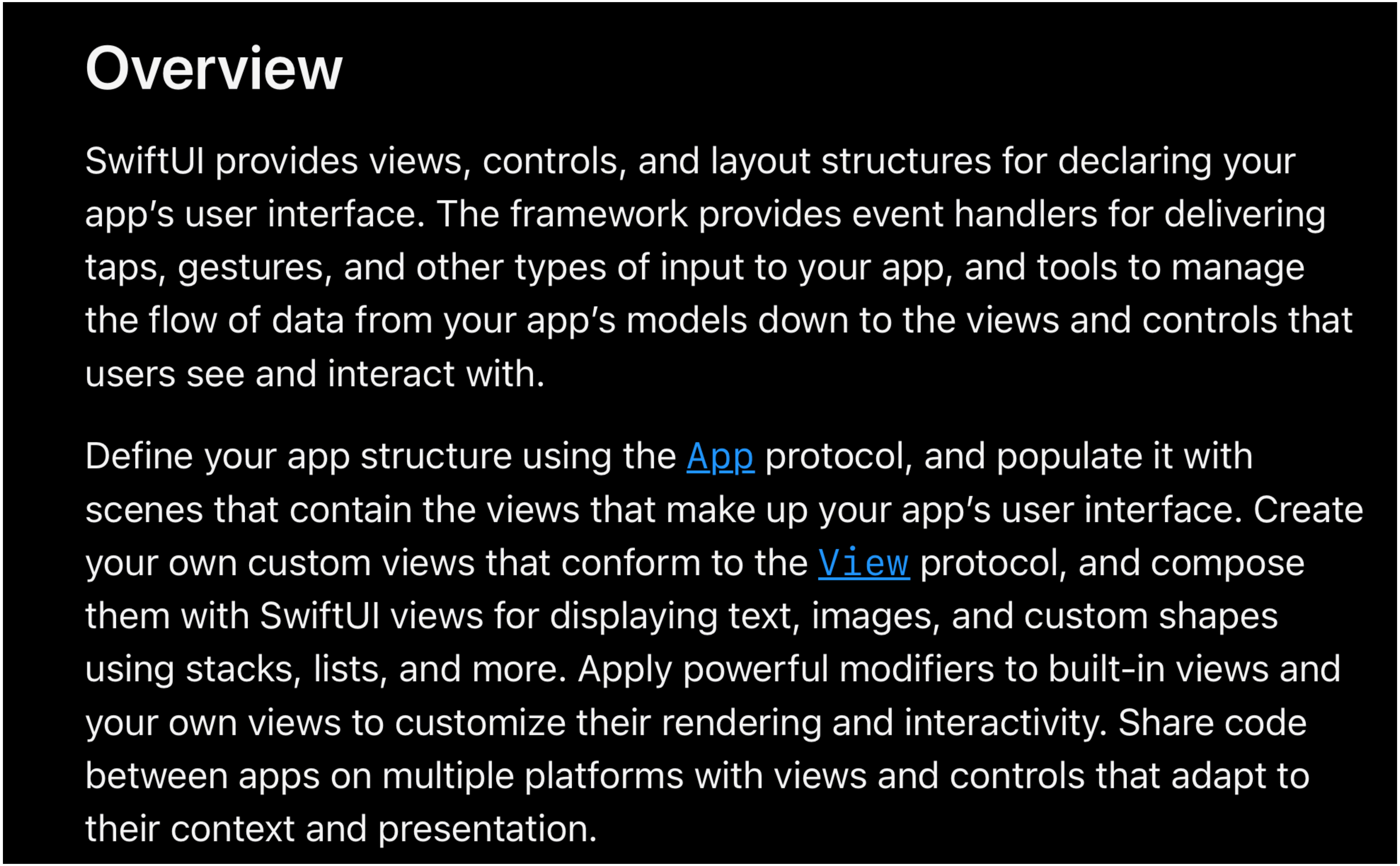


Framework

SwiftUI

Declare the user interface and behavior for your app on every platform.

iOS 13.0+ | iPadOS 13.0+ | Mac Catalyst 13.0+ | macOS 10.15+ | tvOS 13.0+ | visionOS 1.0+ | watchOS 6.0+



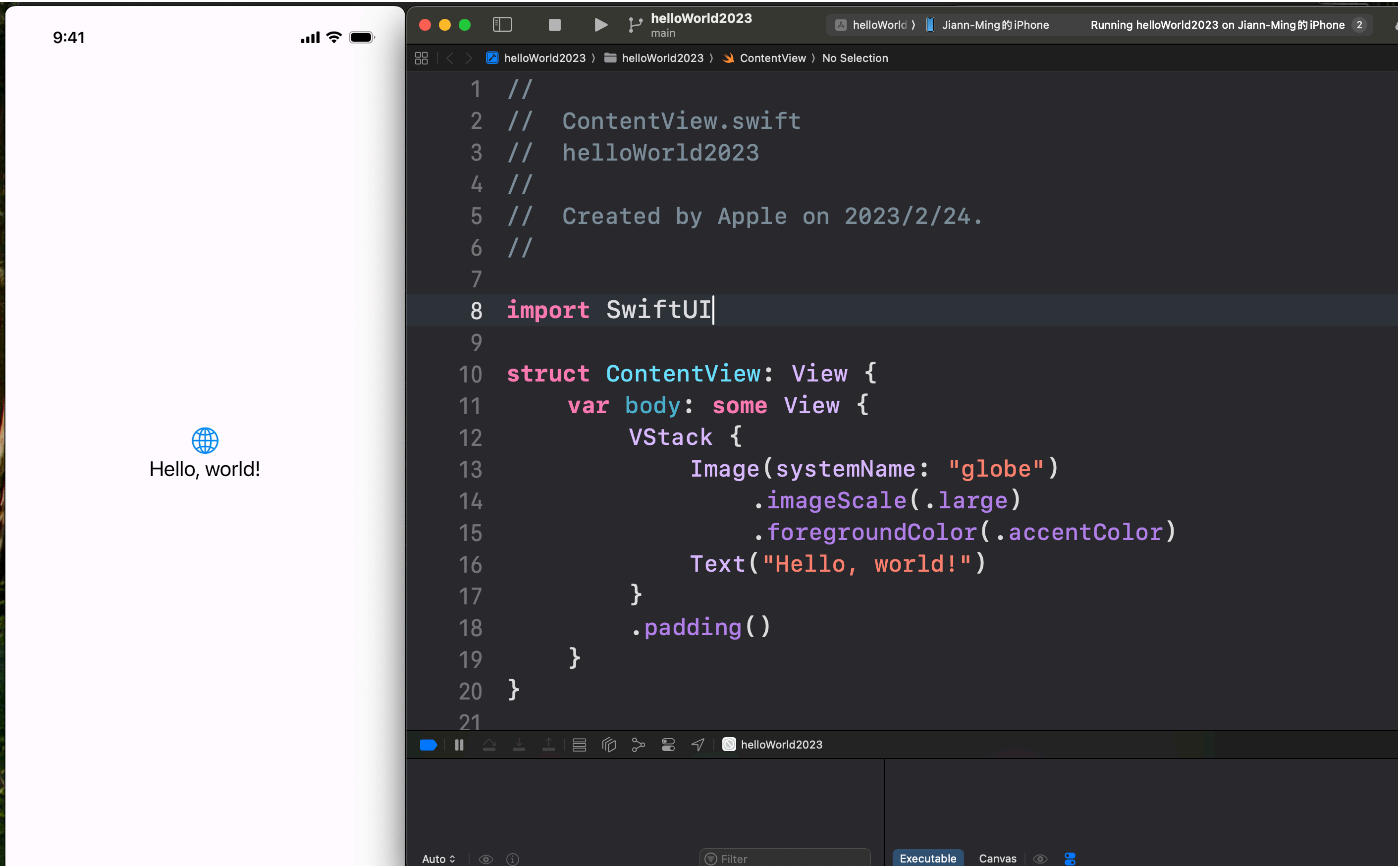
Overview

SwiftUI provides views, controls, and layout structures for declaring your app’s user interface. The framework provides event handlers for delivering taps, gestures, and other types of input to your app, and tools to manage the flow of data from your app’s models down to the views and controls that users see and interact with.

Define your app structure using the `App` protocol, and populate it with scenes that contain the views that make up your app’s user interface. Create your own custom views that conform to the `View` protocol, and compose them with SwiftUI views for displaying text, images, and custom shapes using stacks, lists, and more. Apply powerful modifiers to built-in views and your own views to customize their rendering and interactivity. Share code between apps on multiple platforms with views and controls that adapt to their context and presentation.

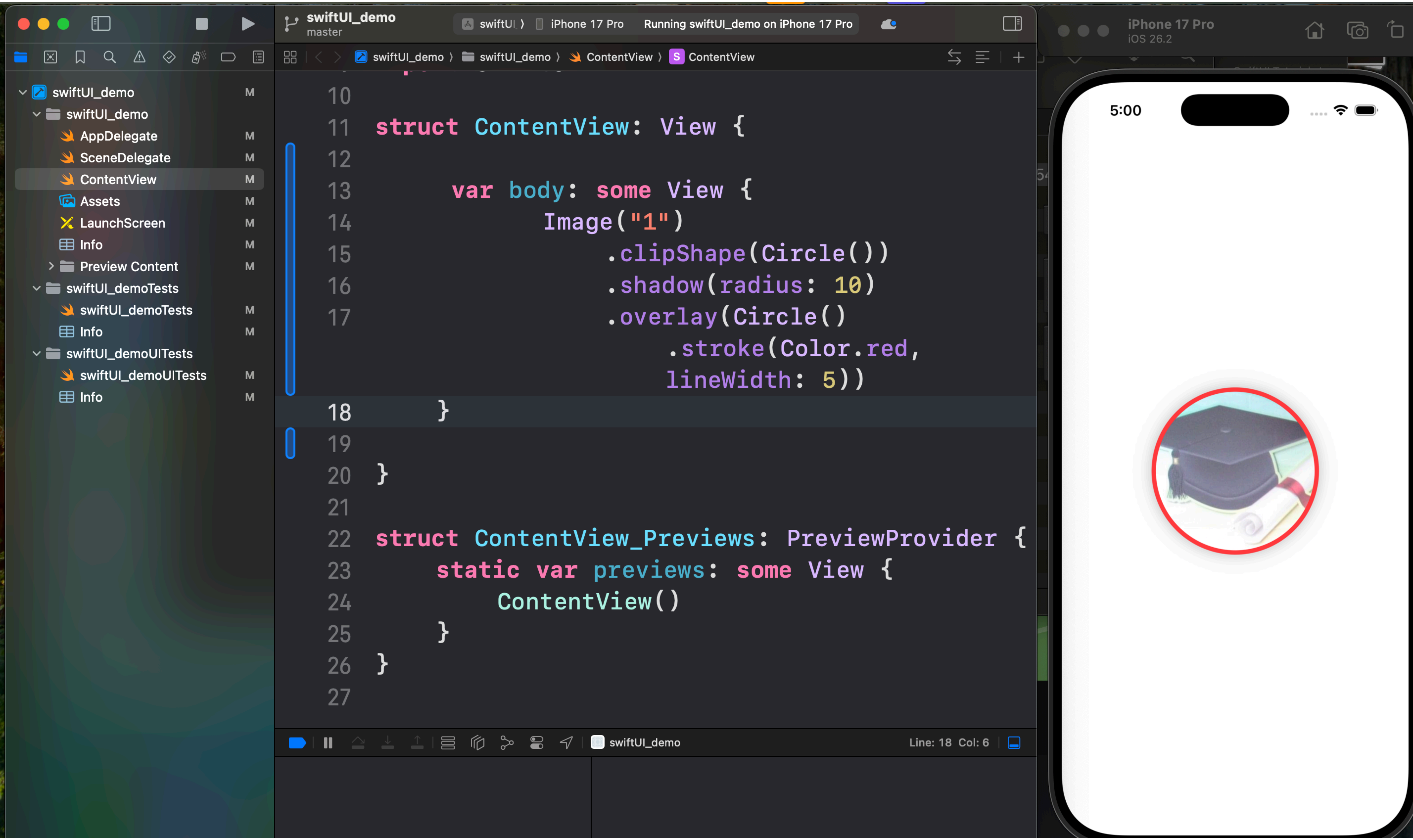
SwiftUI Essentials

Creating and Combining Views



9:41
Hello, world!
helloWorld2023
main
helloWorld 〉 Jiann-Ming的iPhone
Running helloWorld2023 on Jiann-Ming的iPhone 2
helloWorld2023 〉 helloWorld2023 〉 ContentView 〉 No Selection
helloWorld2023
Auto ⌃
Filter
Executable
Canvas

```swift
//
//  ContentView.swift
//  helloWorld2023
//
//  Created by Apple on 2023/2/24.
//

import SwiftUI

struct ContentView: View {
    var body: some View {
        VStack {
            Image(systemName: "globe")
                .imageScale(.large)
                .foregroundColor(.accentColor)
            Text("Hello, world!")
        }
        .padding()
    }
}
```

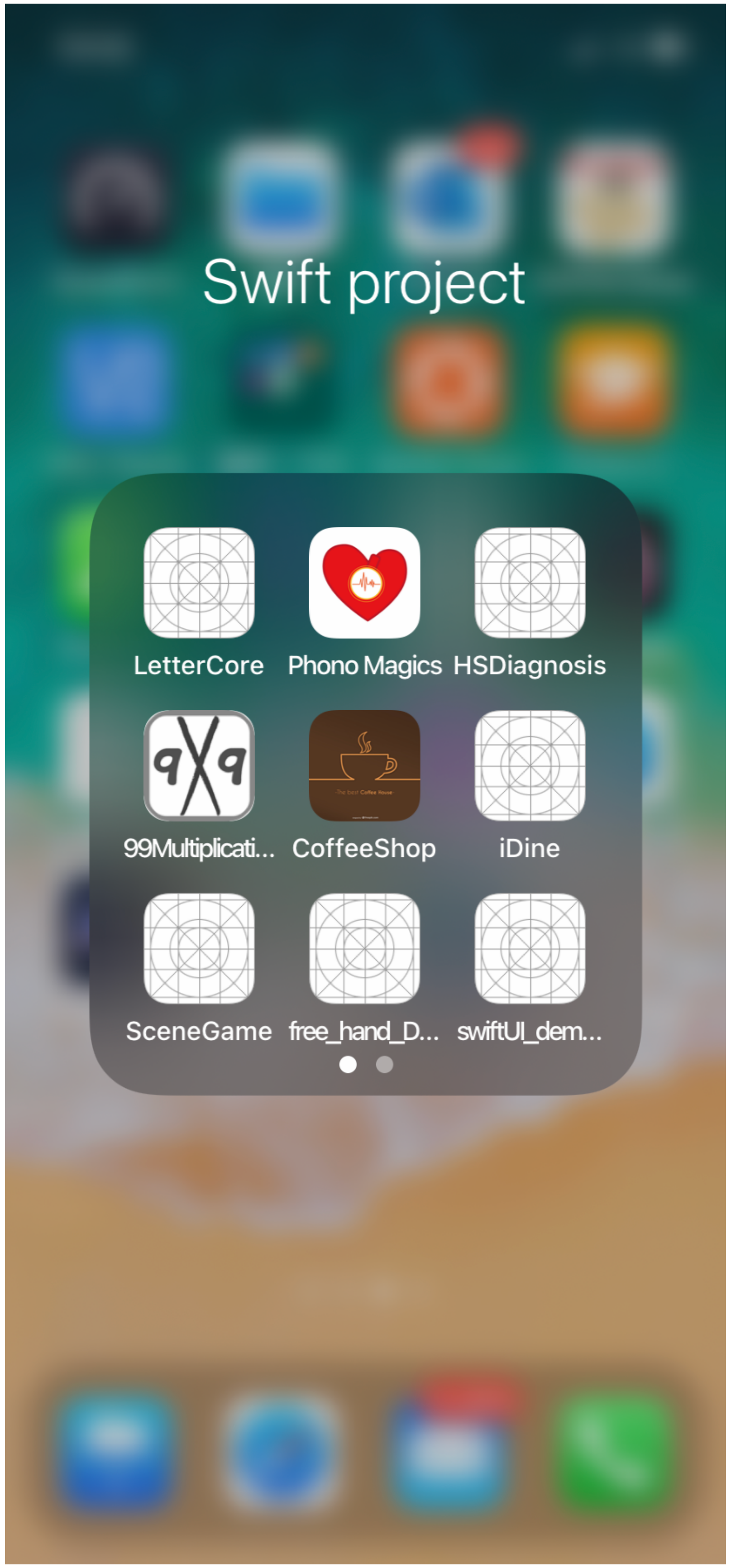


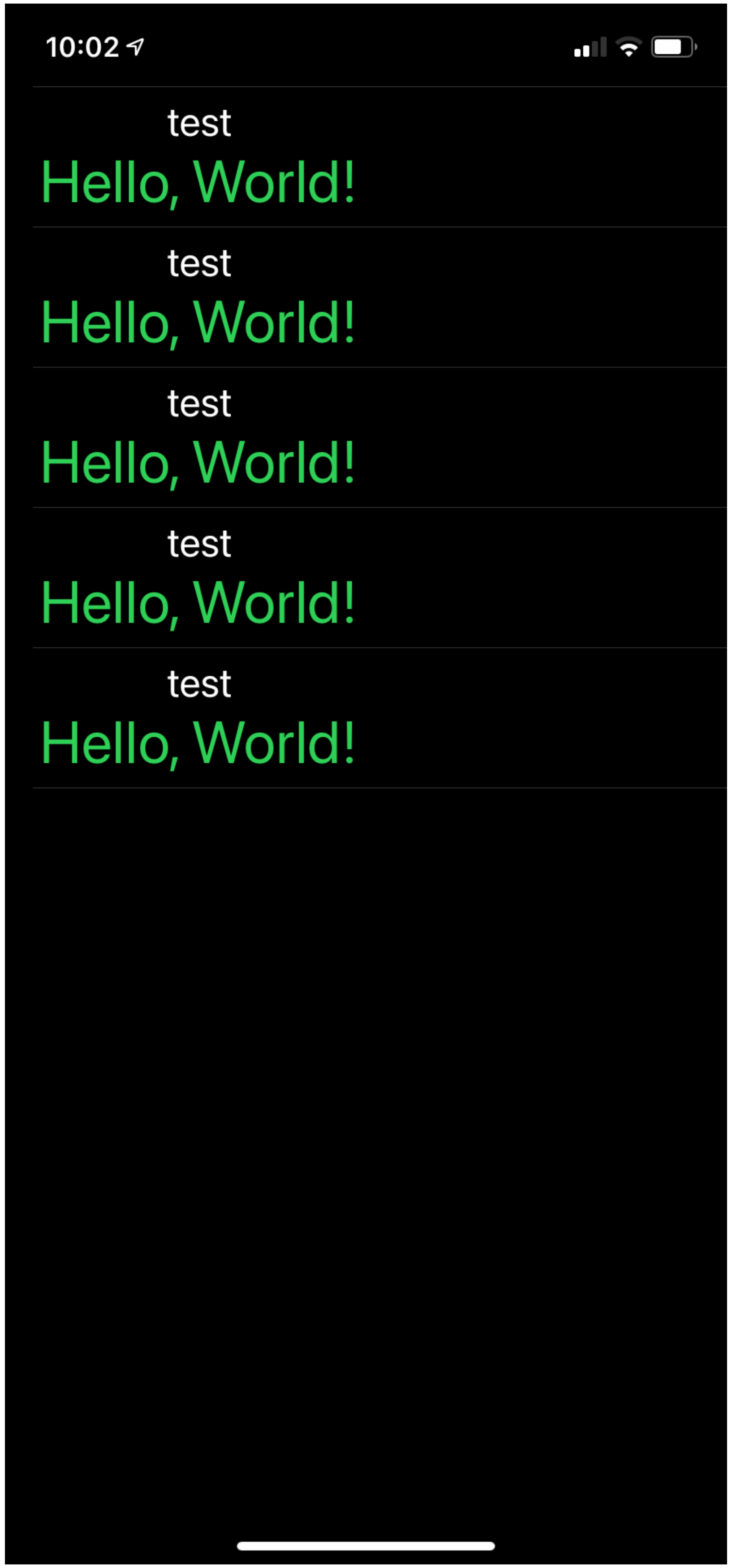
```swift

struct ContentView: View {

    var body: some View {
        Image("1")
            .clipShape(Circle())
            .shadow(radius: 10)
            .overlay(Circle()
                .stroke(Color.red,
                lineWidth: 5))
    }

}

struct ContentView_Previews: PreviewProvider {
    static var previews: some View {
        ContentView()
    }
}

```

Hello world

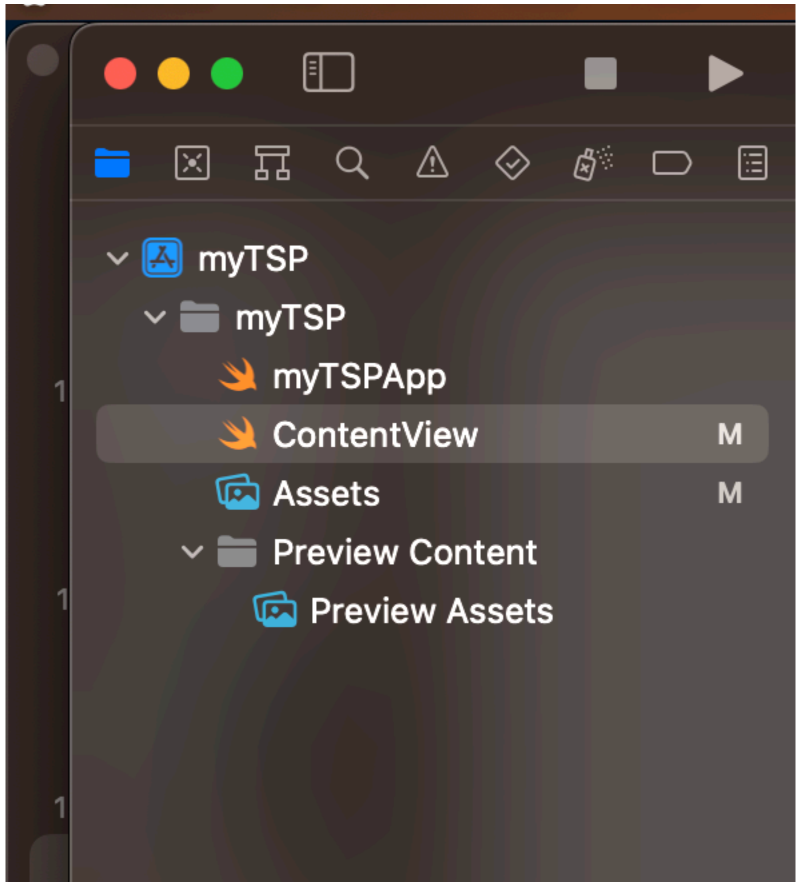
Display texts

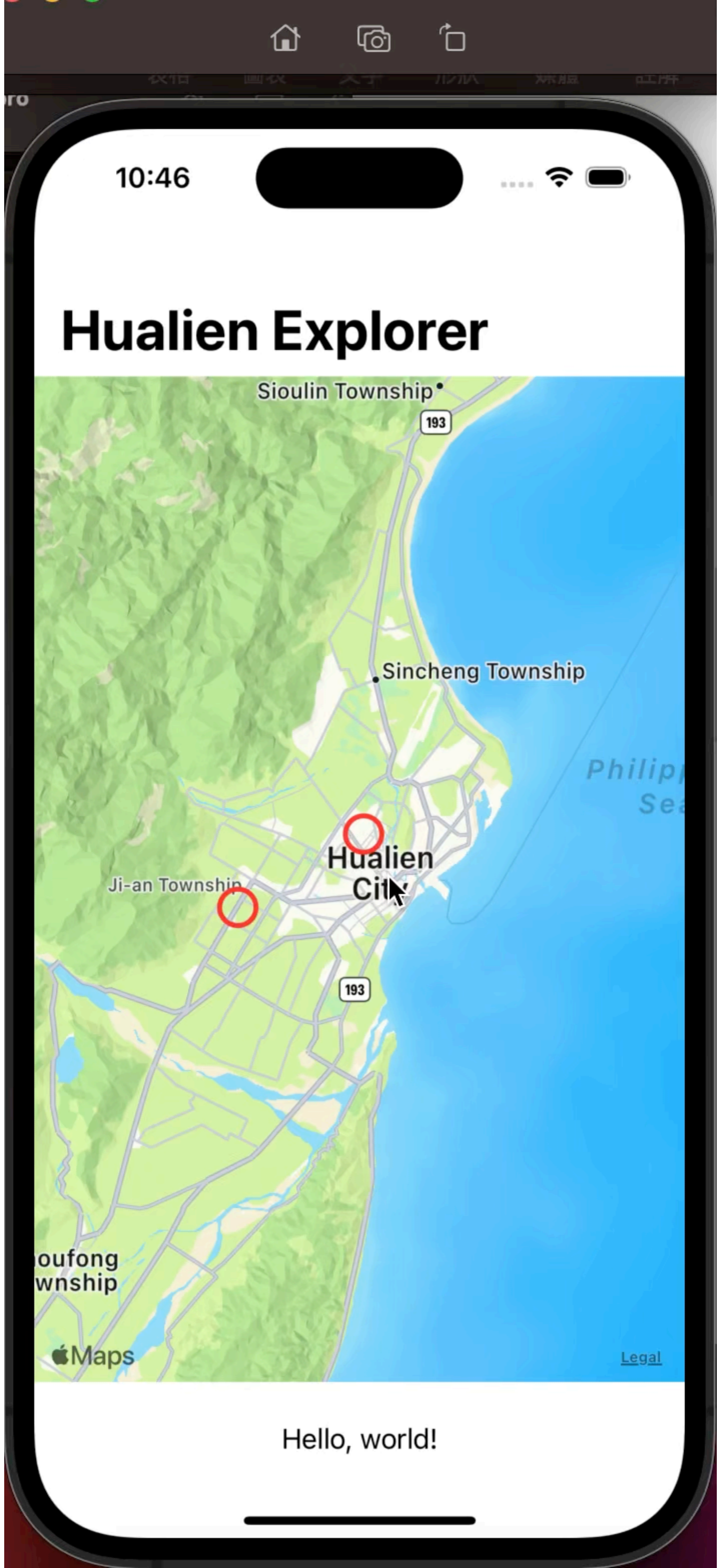
iteratively





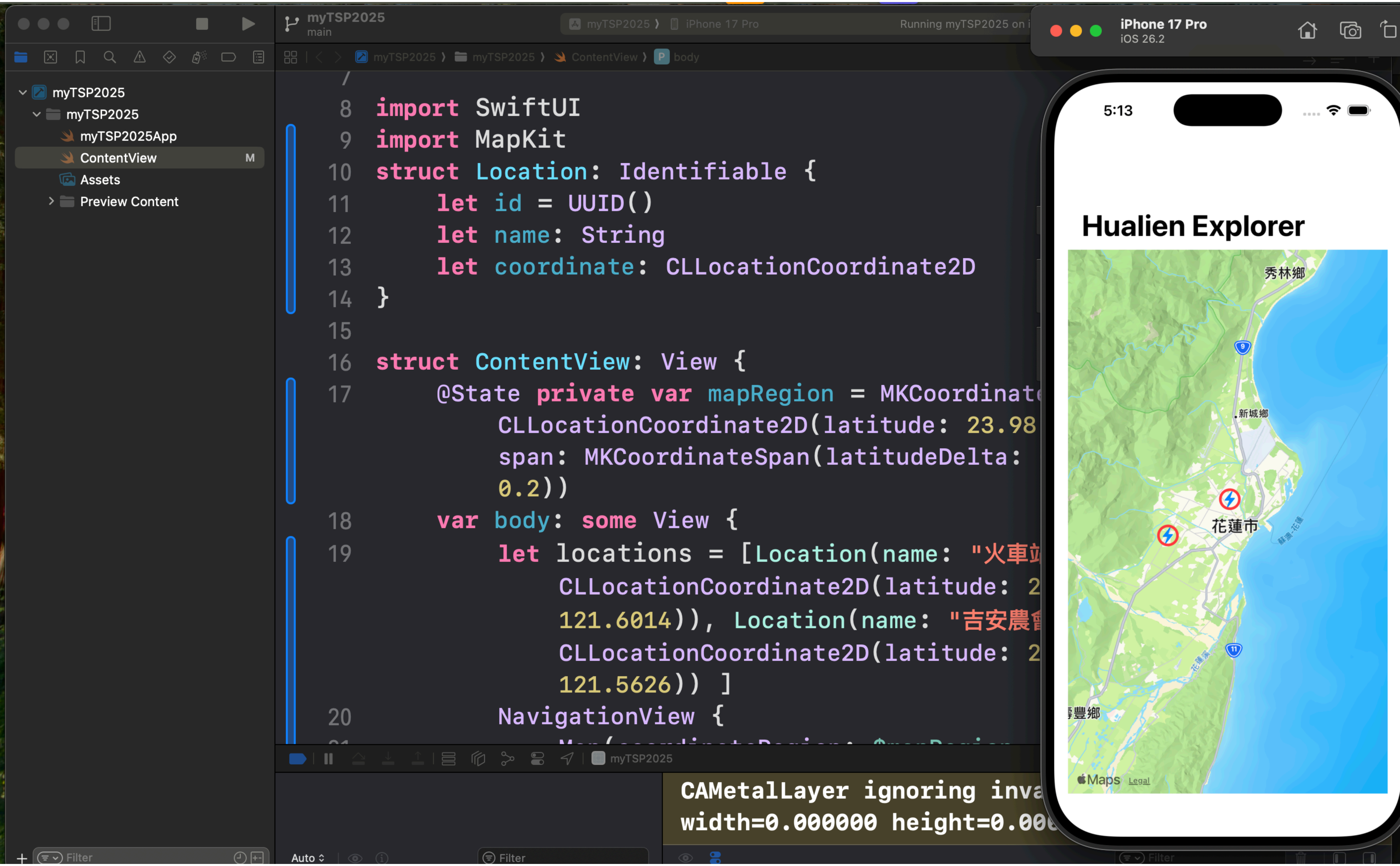
```
test
Hello, World!
test
Hello, World!
test
Hello, World!
test
Hello, World!
test
Hello, World!
```





Hualien Explorer

Hello, world!



```
import SwiftUI
import MapKit
struct Location: Identifiable {
    let id = UUID()
    let name: String
    let coordinate: CLLocationCoordinate2D
}

struct ContentView: View {
    @State private var mapRegion = MKCoordinat
        CLLocationCoordinate2D(latitude: 23.98
        span: MKCoordinateSpan(latitudeDelta:
        0.2))
    var body: some View {
        let locations = [Location(name: "火車
            CLLocationCoordinate2D(latitude: 2
            121.6014)), Location(name: "吉安農
            CLLocationCoordinate2D(latitude: 2
            121.5626)) ]
        NavigationView {
```

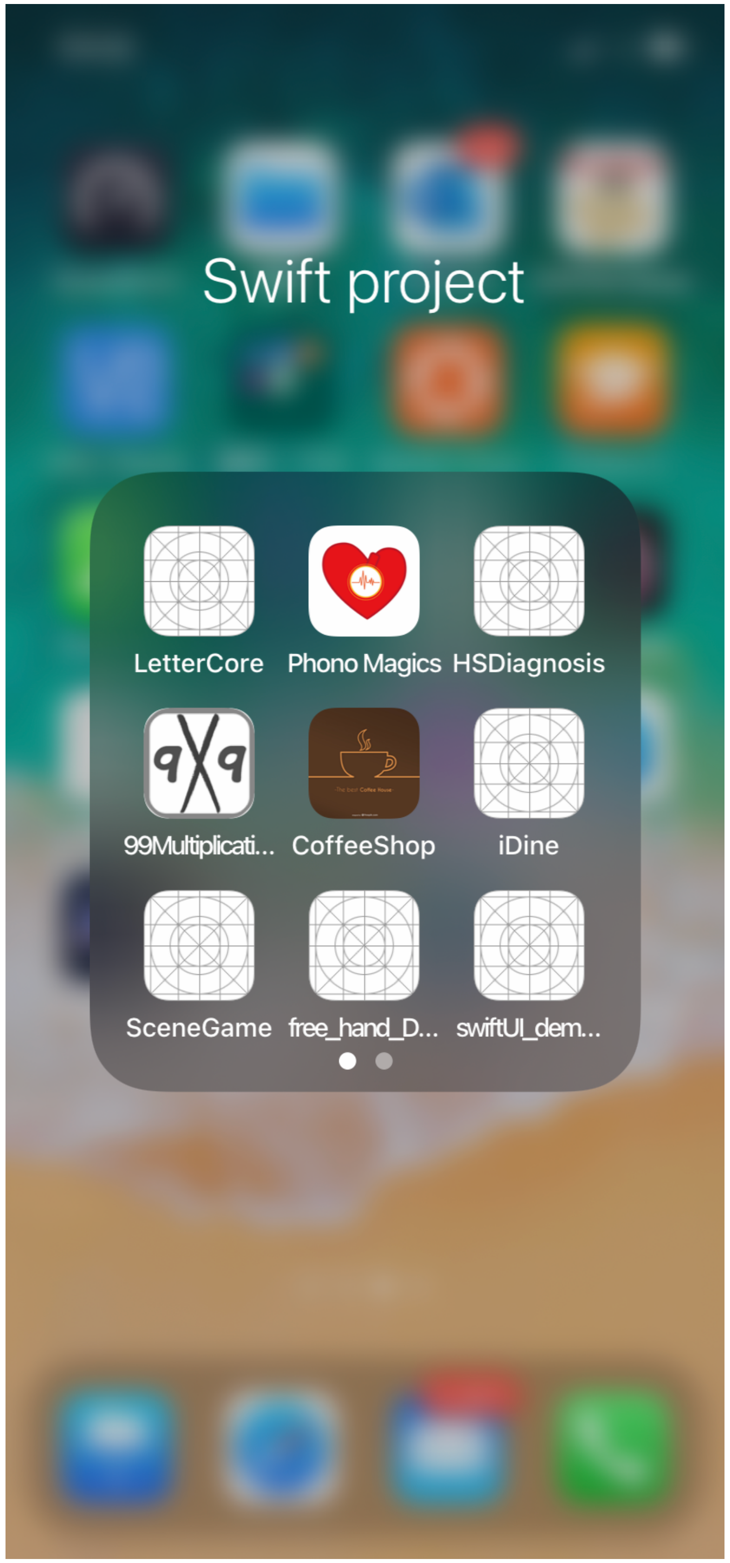
```
CAMetalLayer ignoring inva
width=0.000000 height=0.00
```

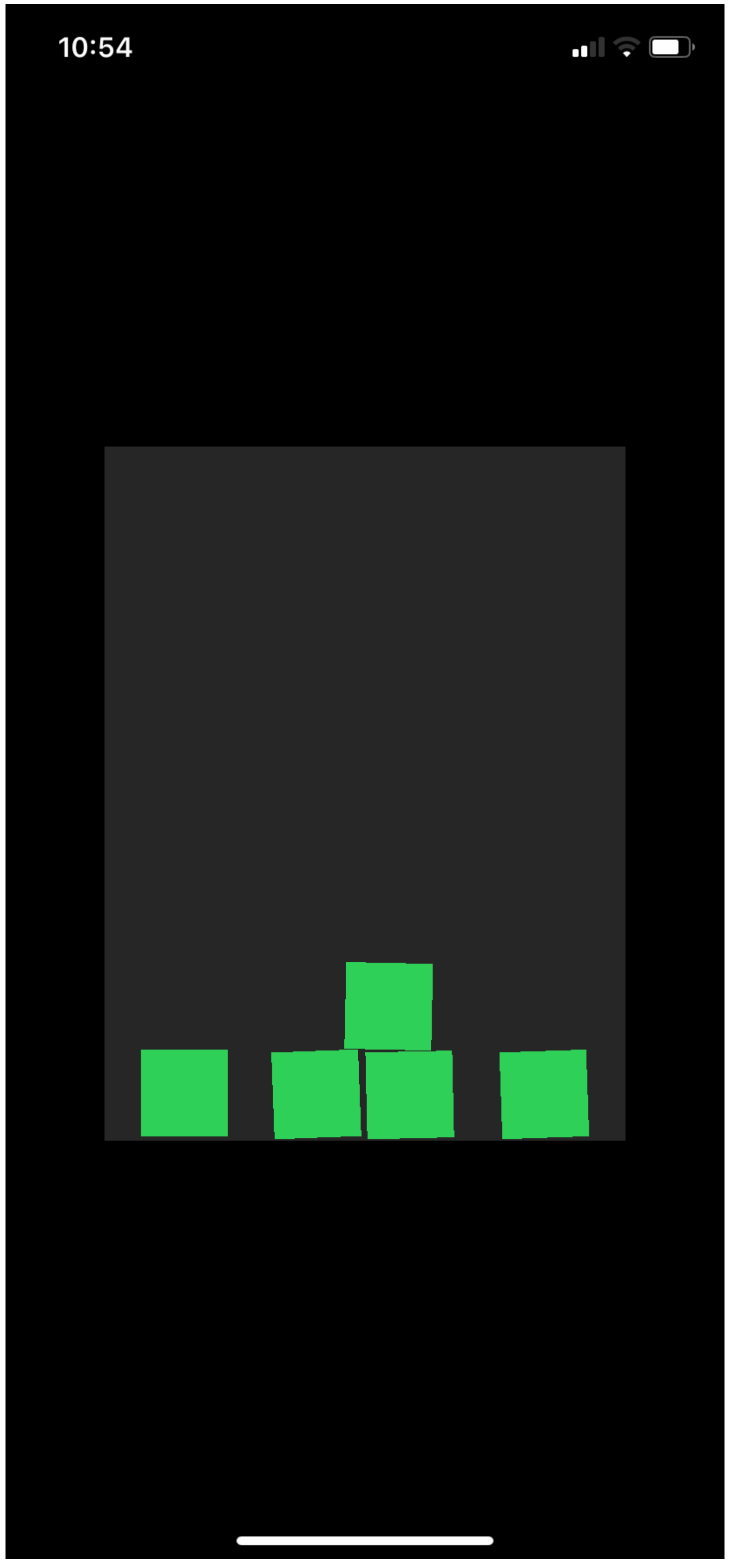
Hualien Explorer

SceneGame

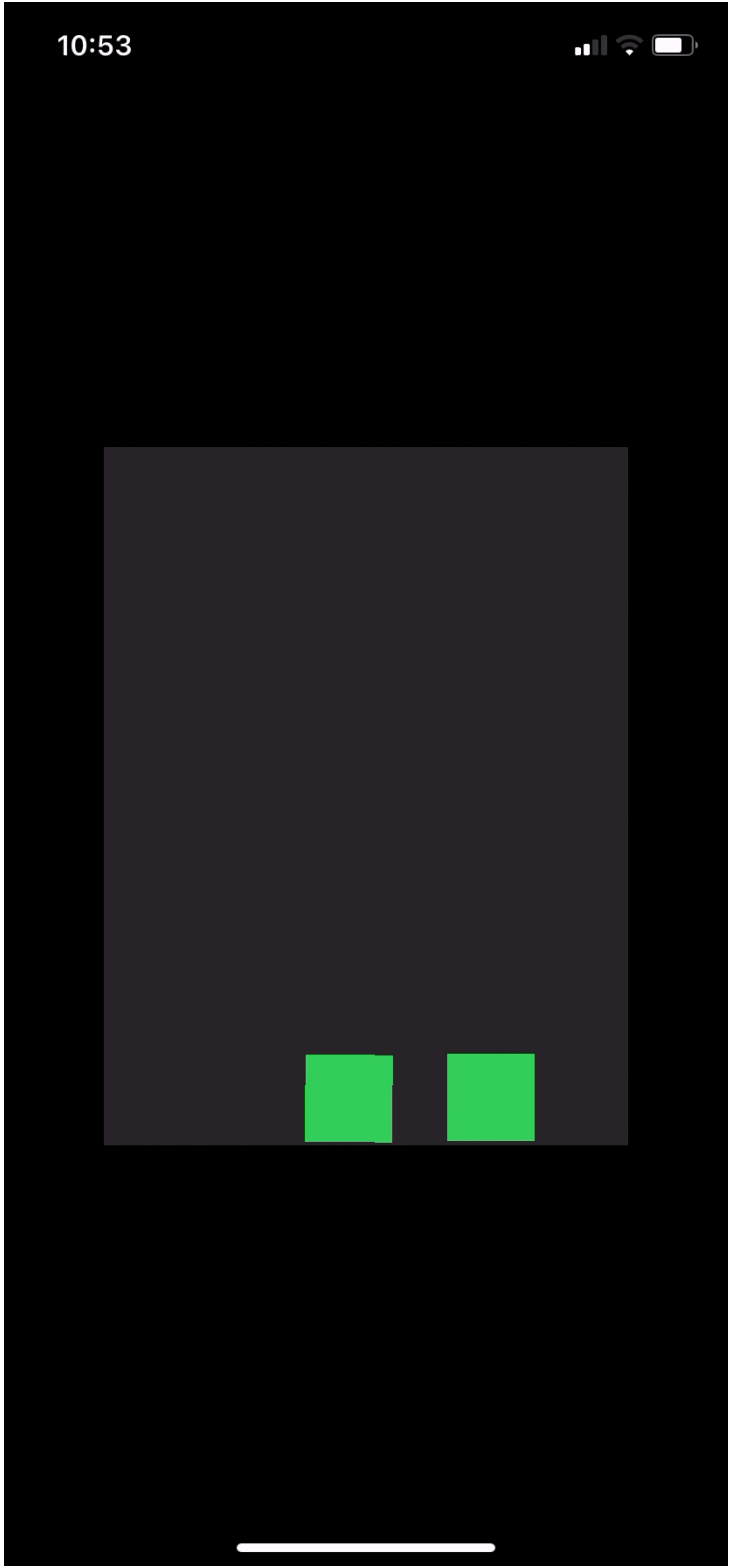
A scene game

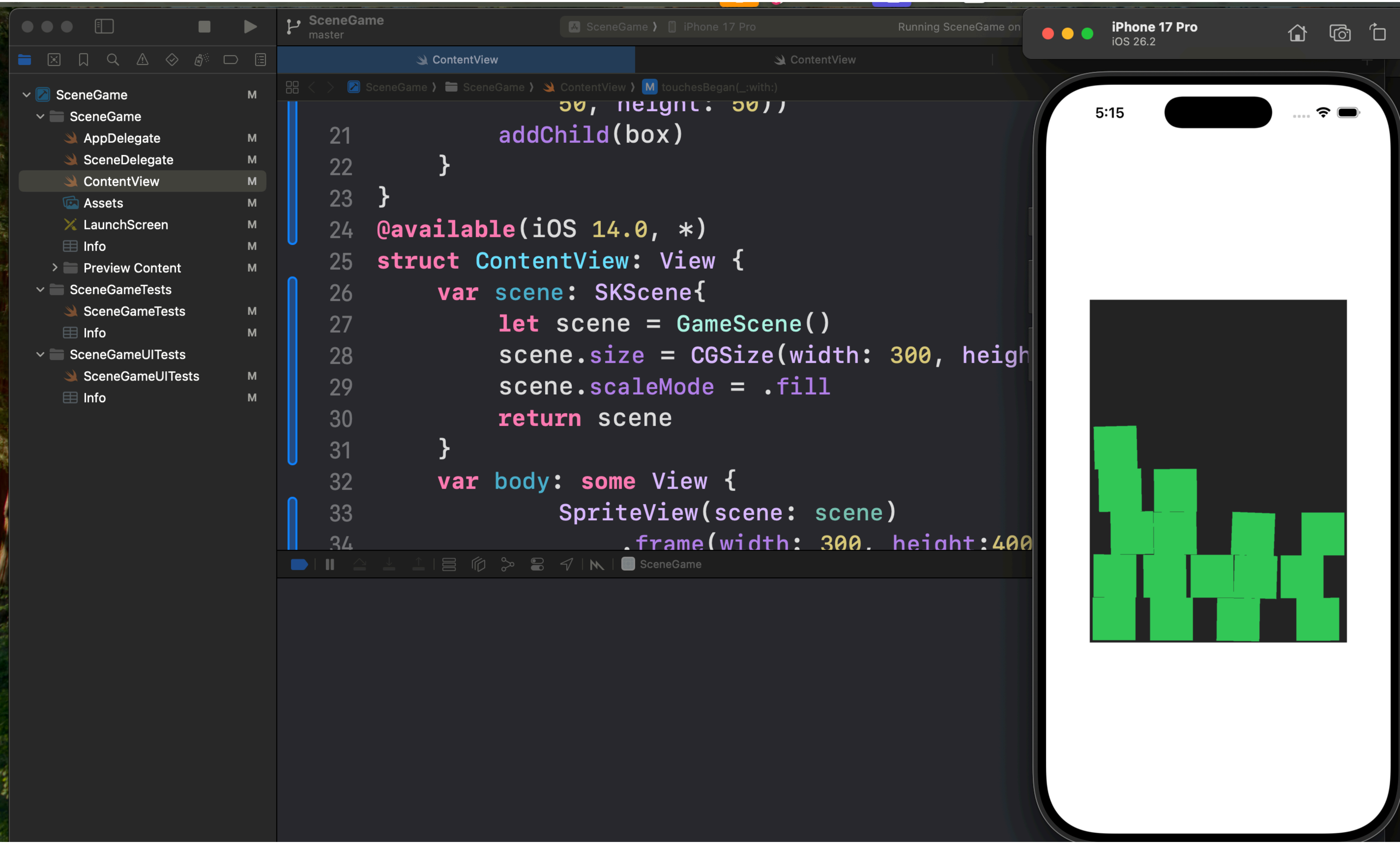
Animation of blocks





SceneGame

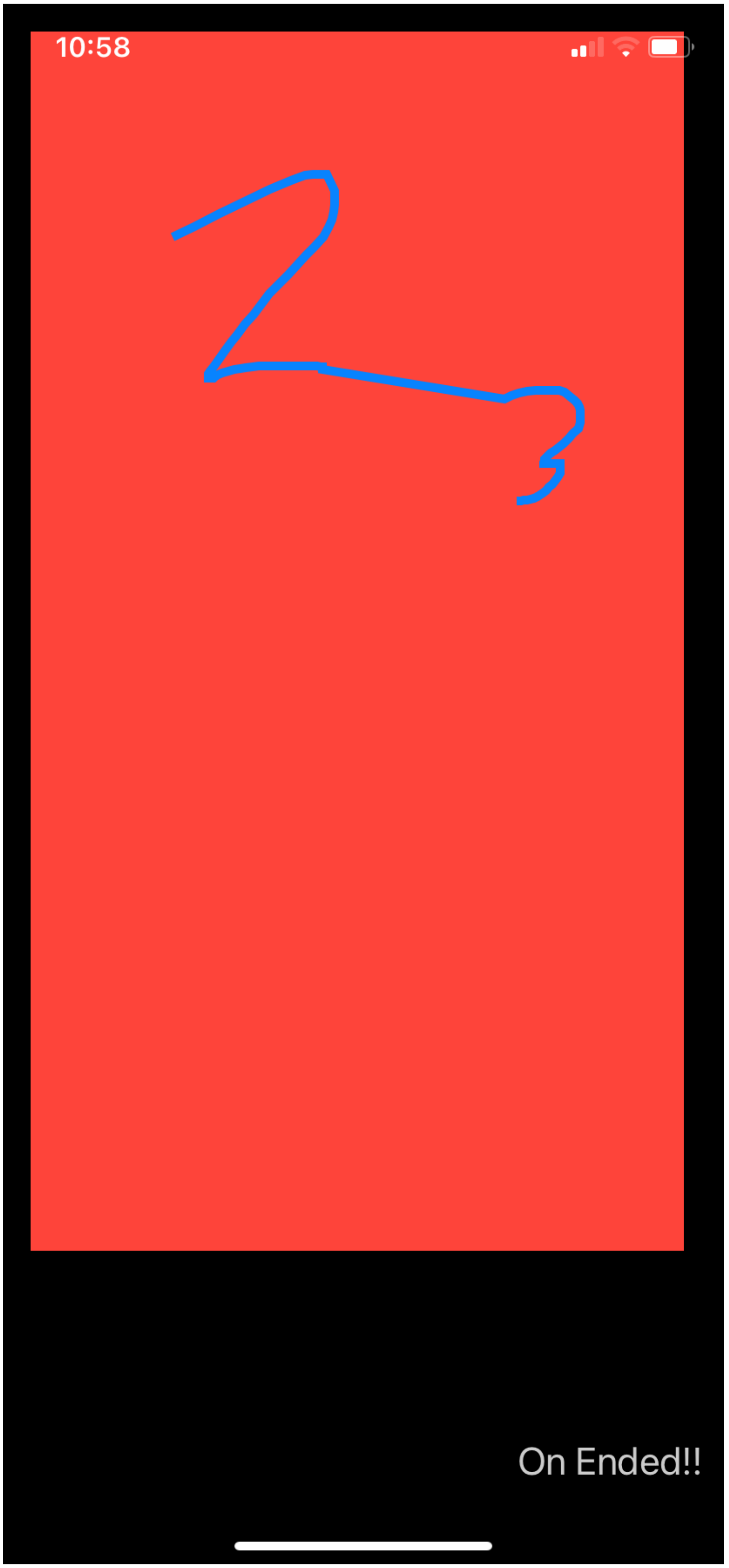


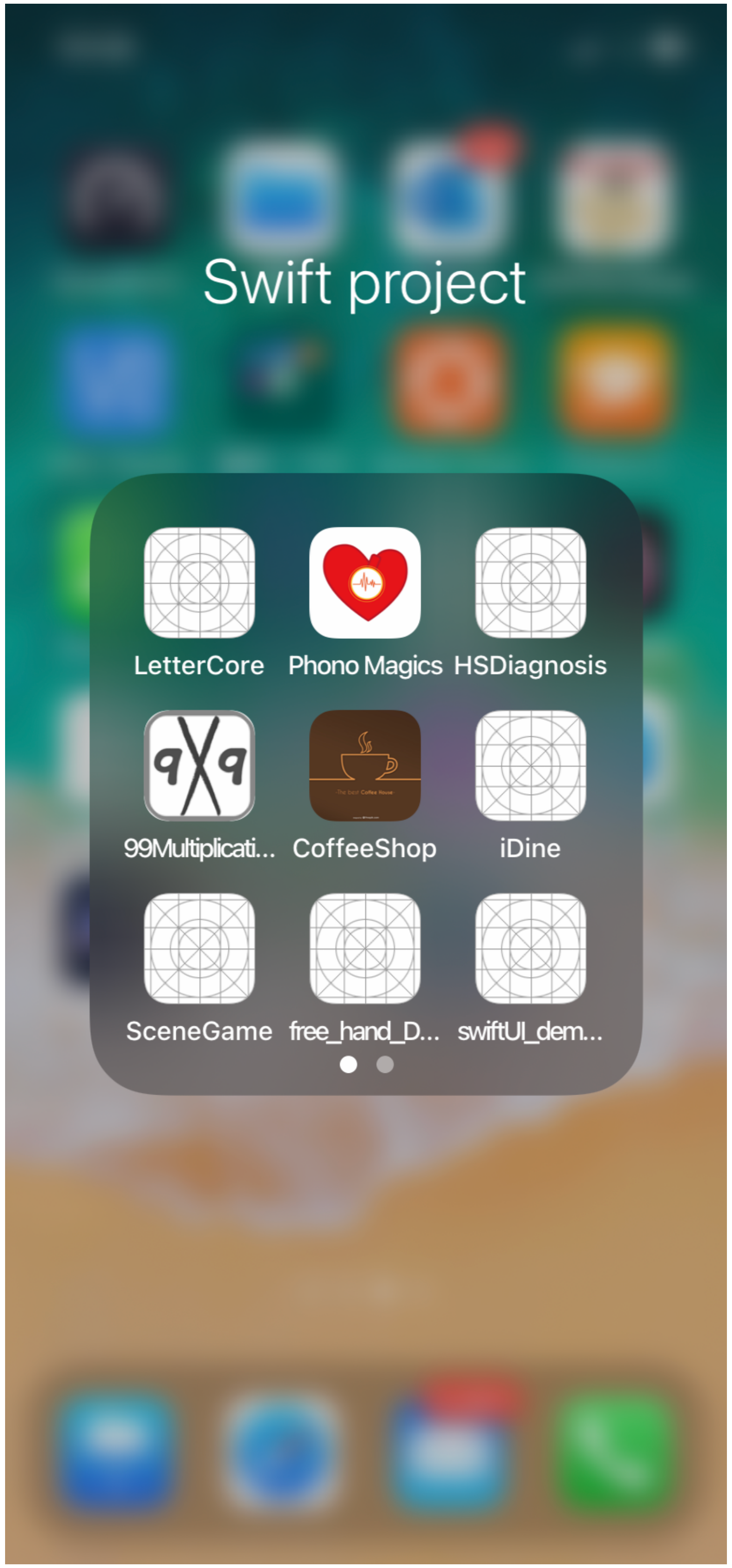


```
            50, height: 50))
        addChild(box)
    }
}
@available(iOS 14.0, *)
struct ContentView: View {
    var scene: SKScene{
        let scene = GameScene()
        scene.size = CGSize(width: 300, heigh
        scene.scaleMode = .fill
        return scene
    }
    var body: some View {
            SpriteView(scene: scene)
                .frame(width: 300, height:400
```

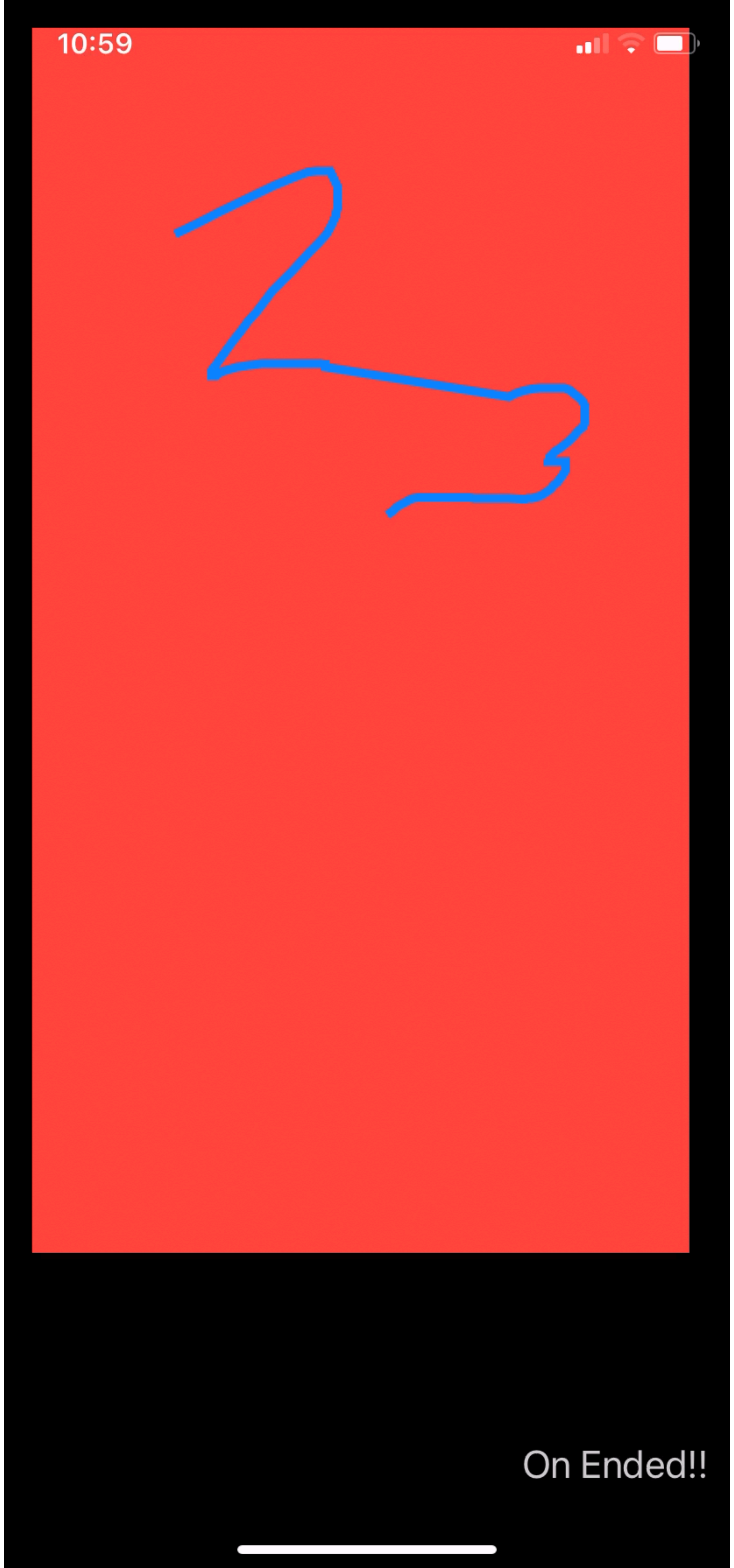
Free Hand Draw

- Track gesture
- Free hand draw on touch screen



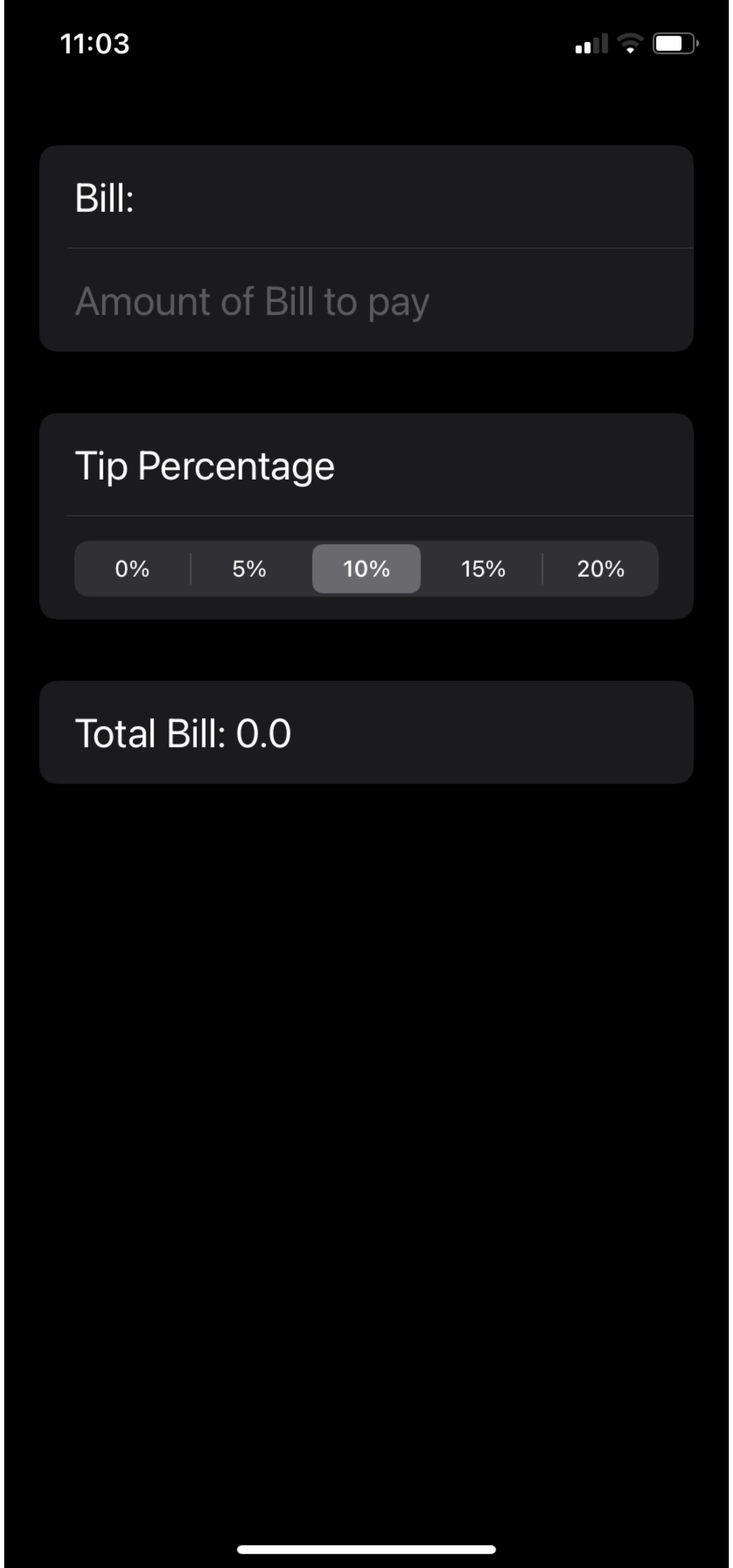


Free Hand Draw



Tip Calculator

- Get bill
- Picker
- Calculate amount



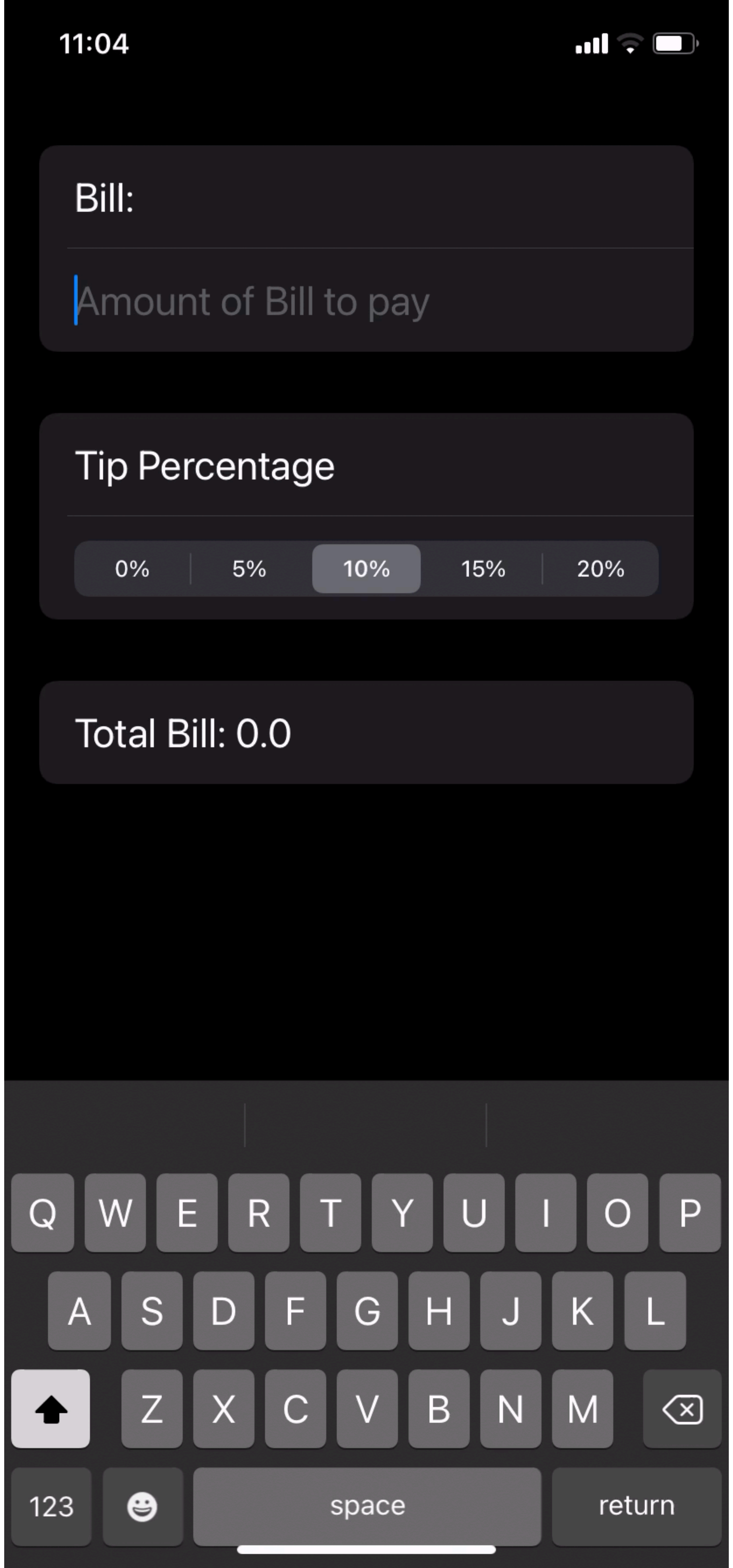
Bill:

Amount of Bill to pay

Tip Percentage

Total Bill: 0.0

Tip Calculator



Bill:

Amount of Bill to pay

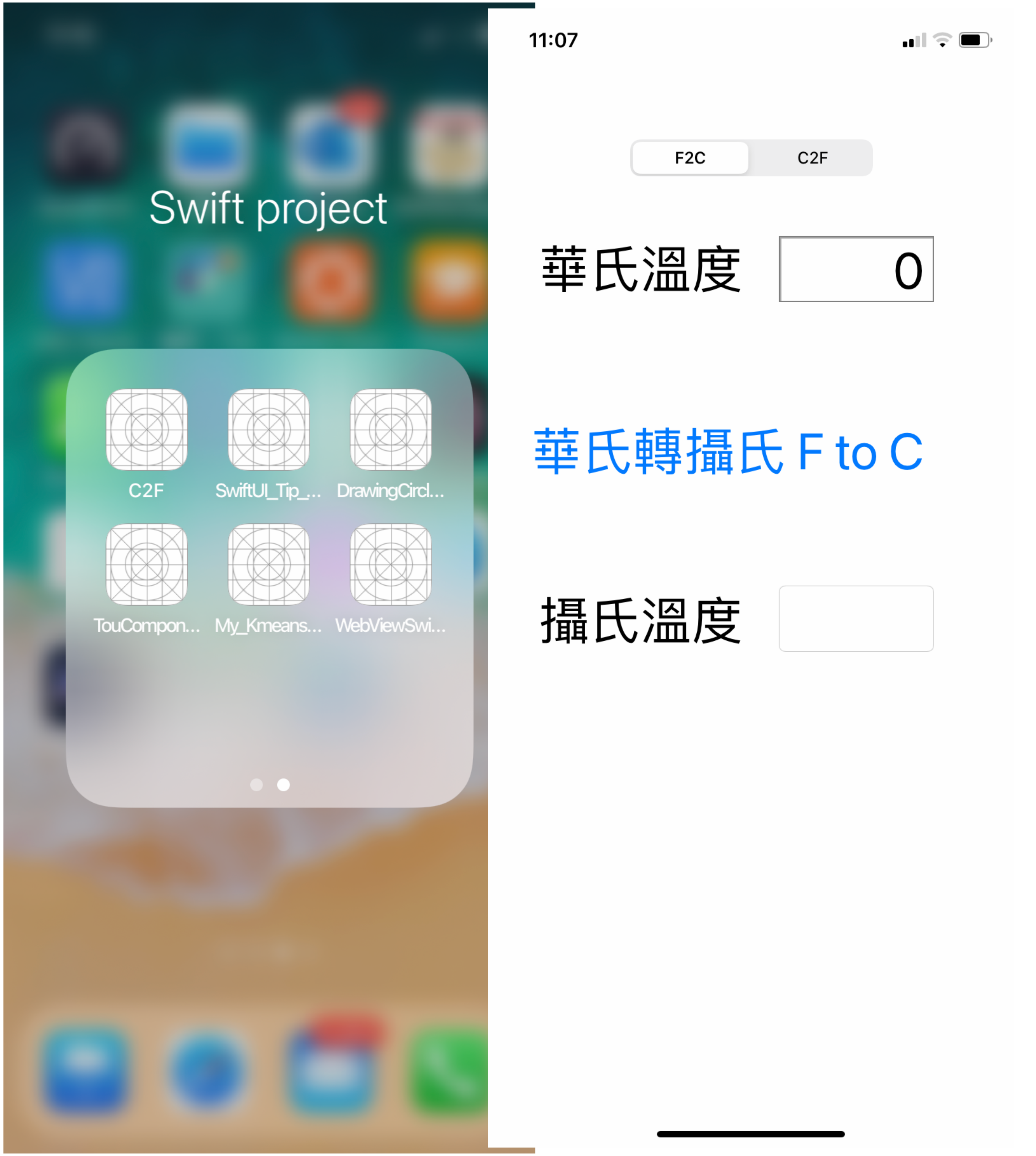
Tip Percentage

0% | 5% | [✓ 10%] | 15% | 20%

Total Bill: 0.0

C2F

Temperature transferred



11:07

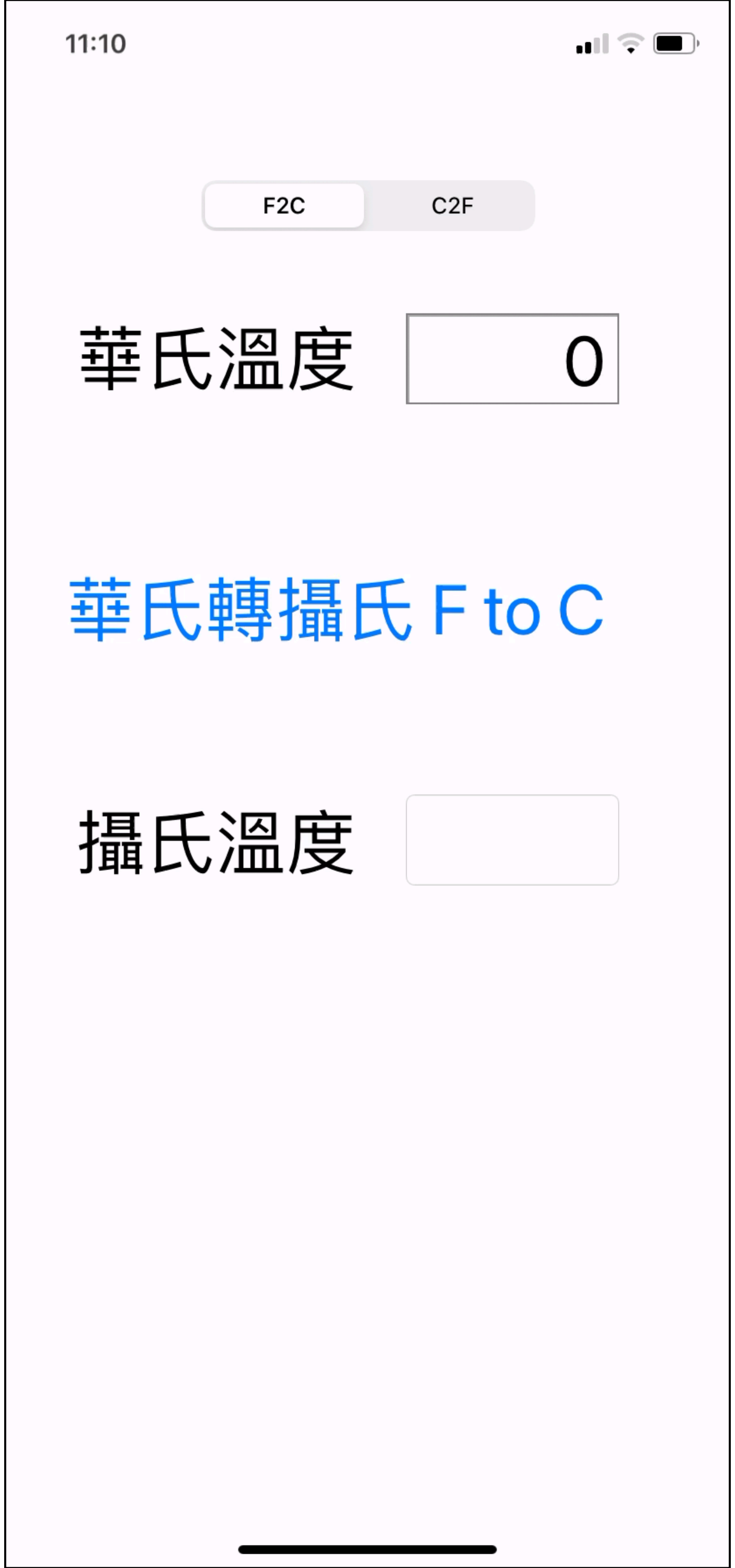
F2C C2F

華氏溫度 0

華氏轉攝氏 F to C

攝氏溫度

C2F



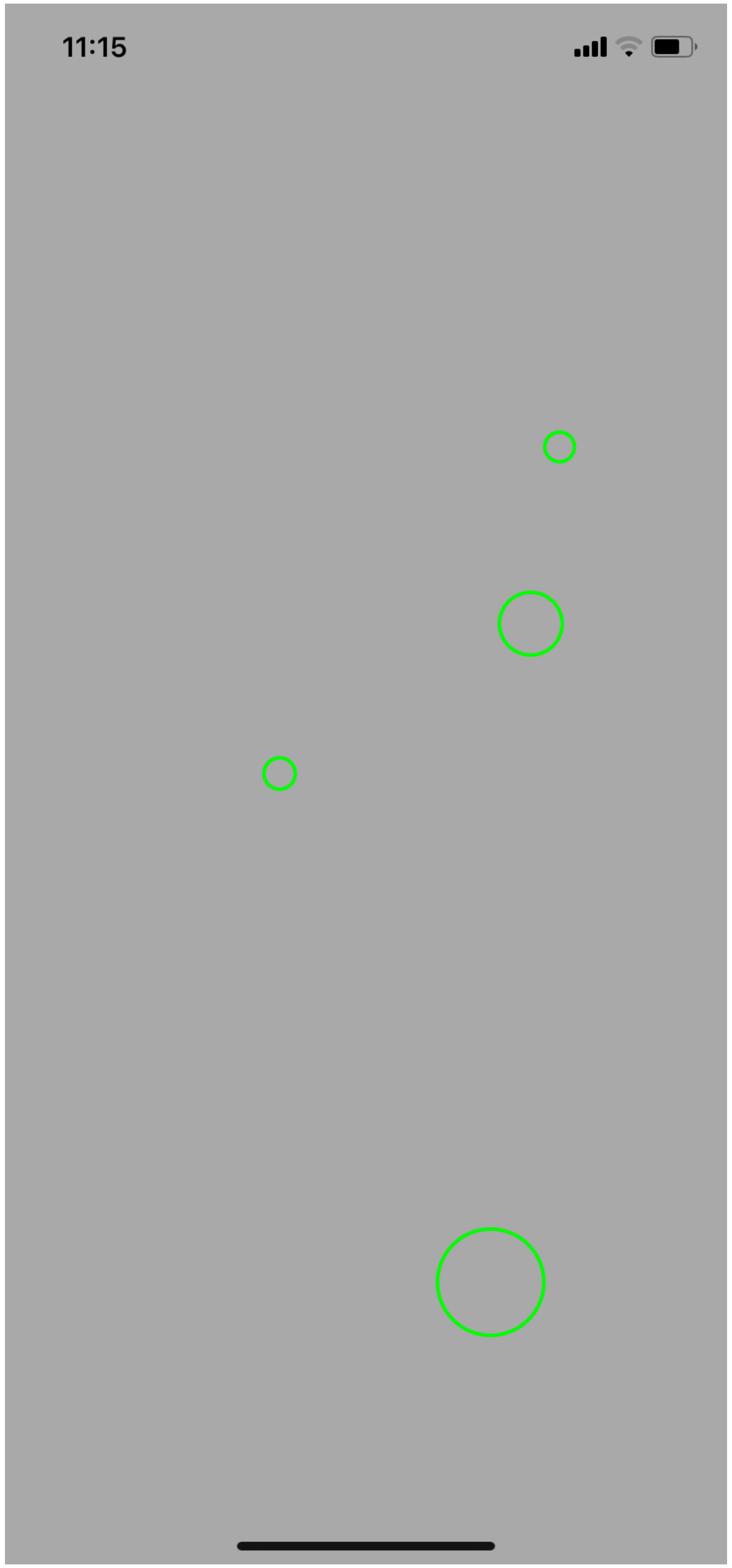
華氏溫度 0

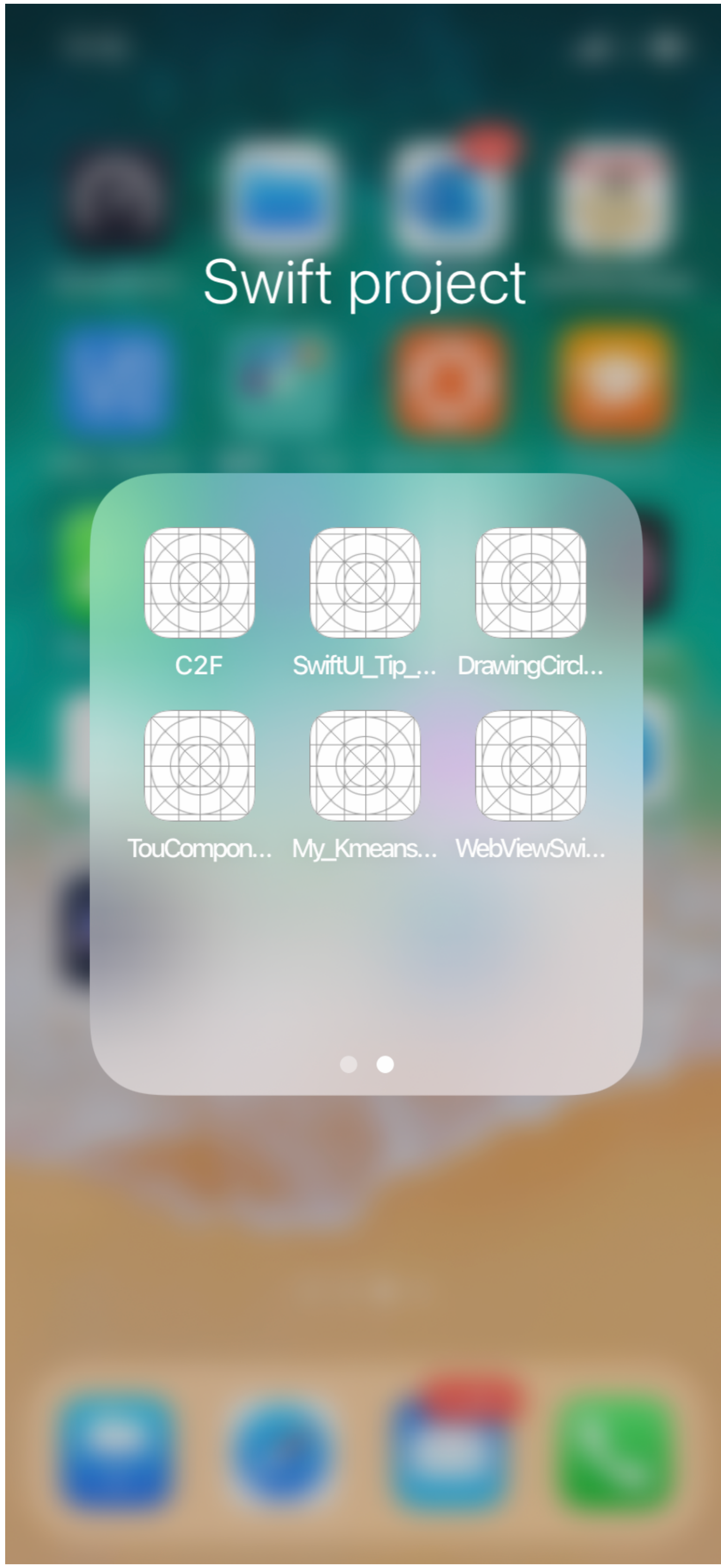
華氏轉攝氏 F to C

攝氏溫度

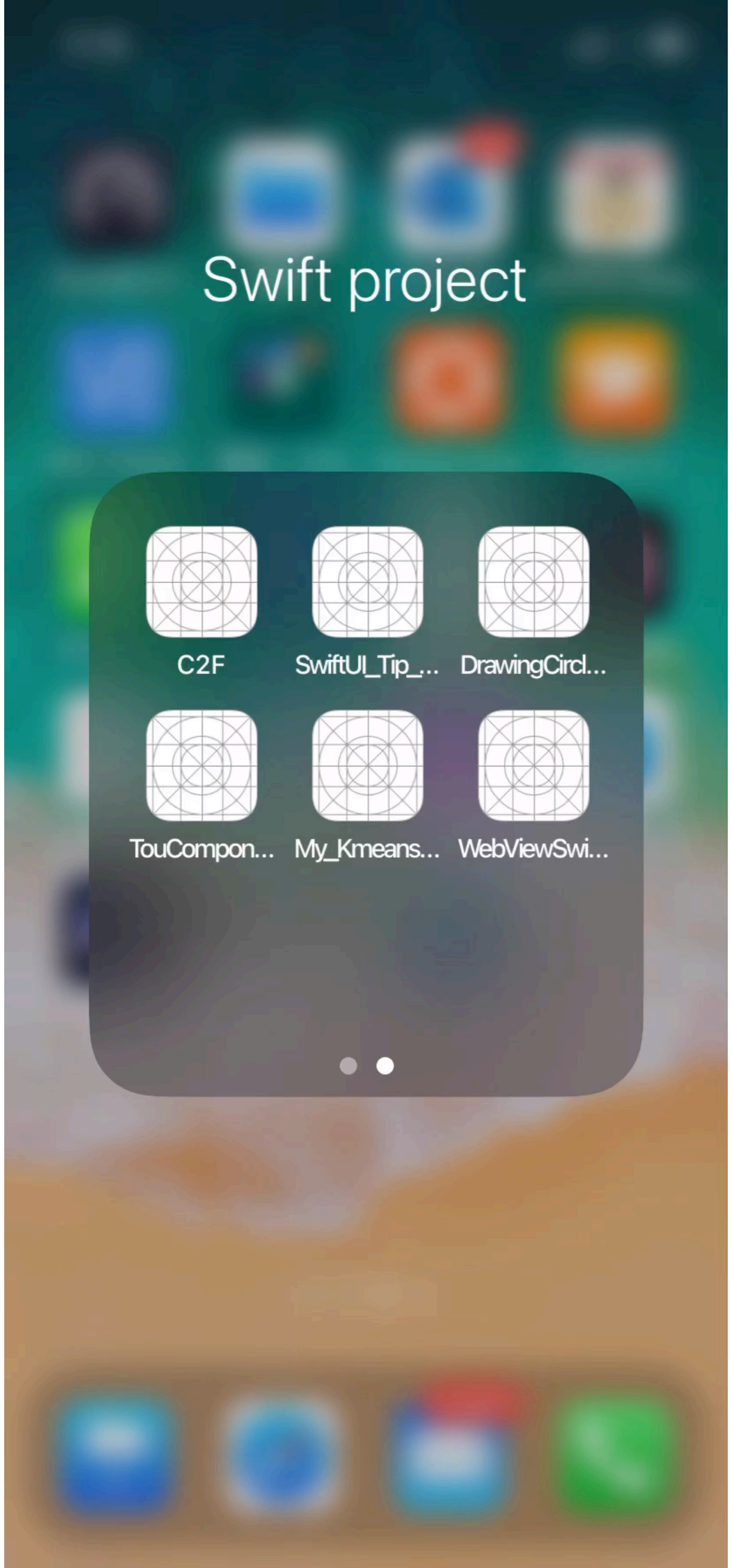
Draw Circles

- Draw a circle at where finger touches screen



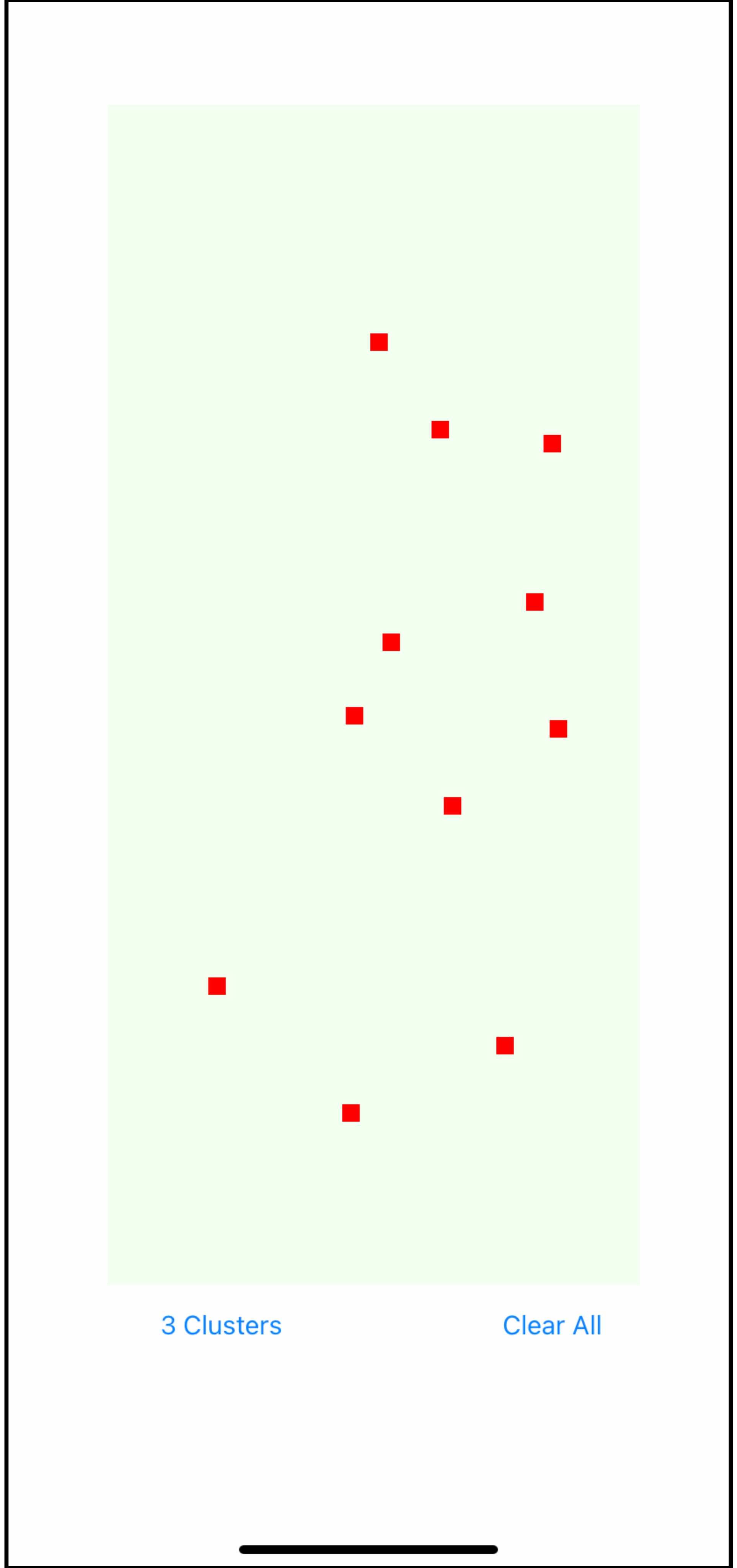


Draw Circles

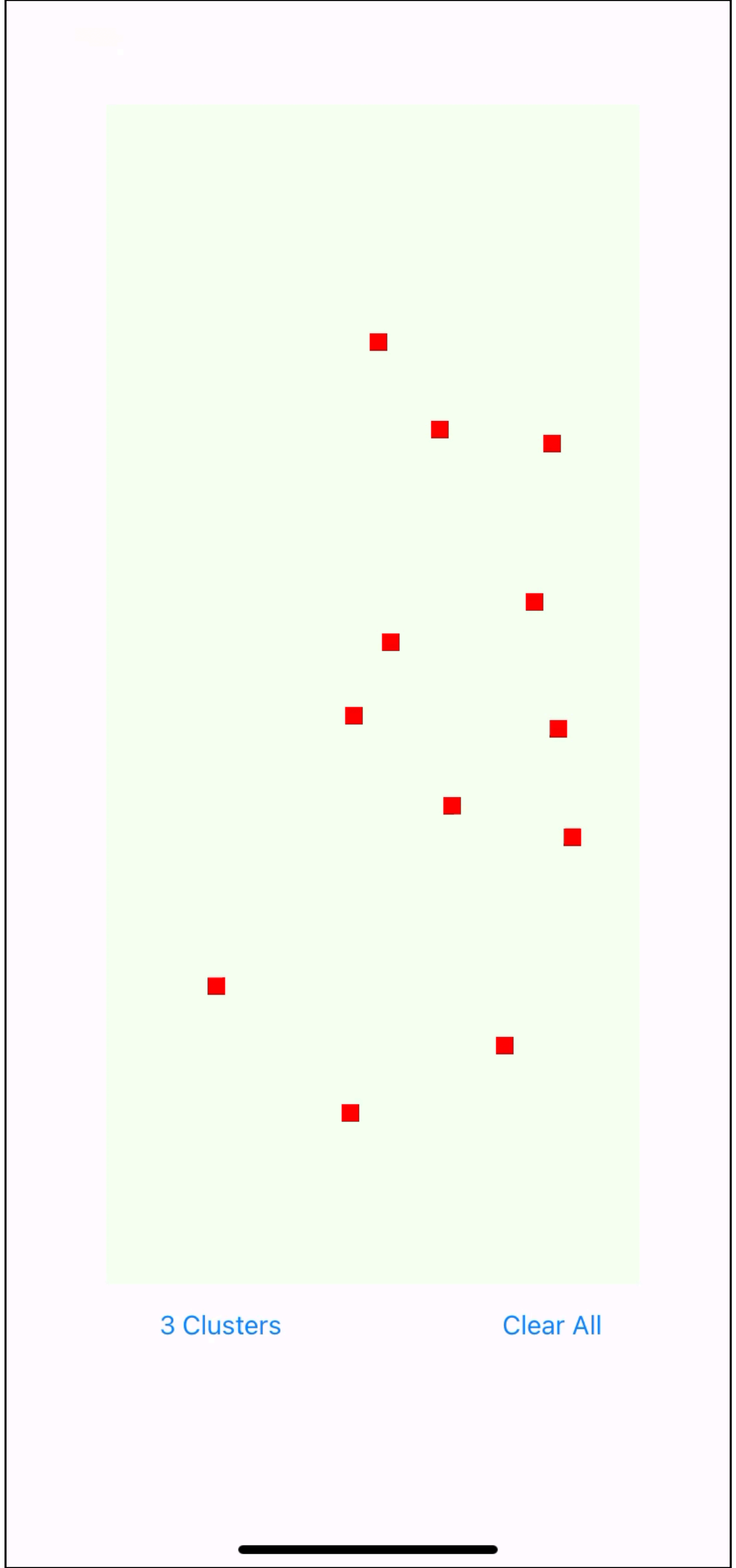


Clustering

- Touch screen to place points
- Display points
- Find centers

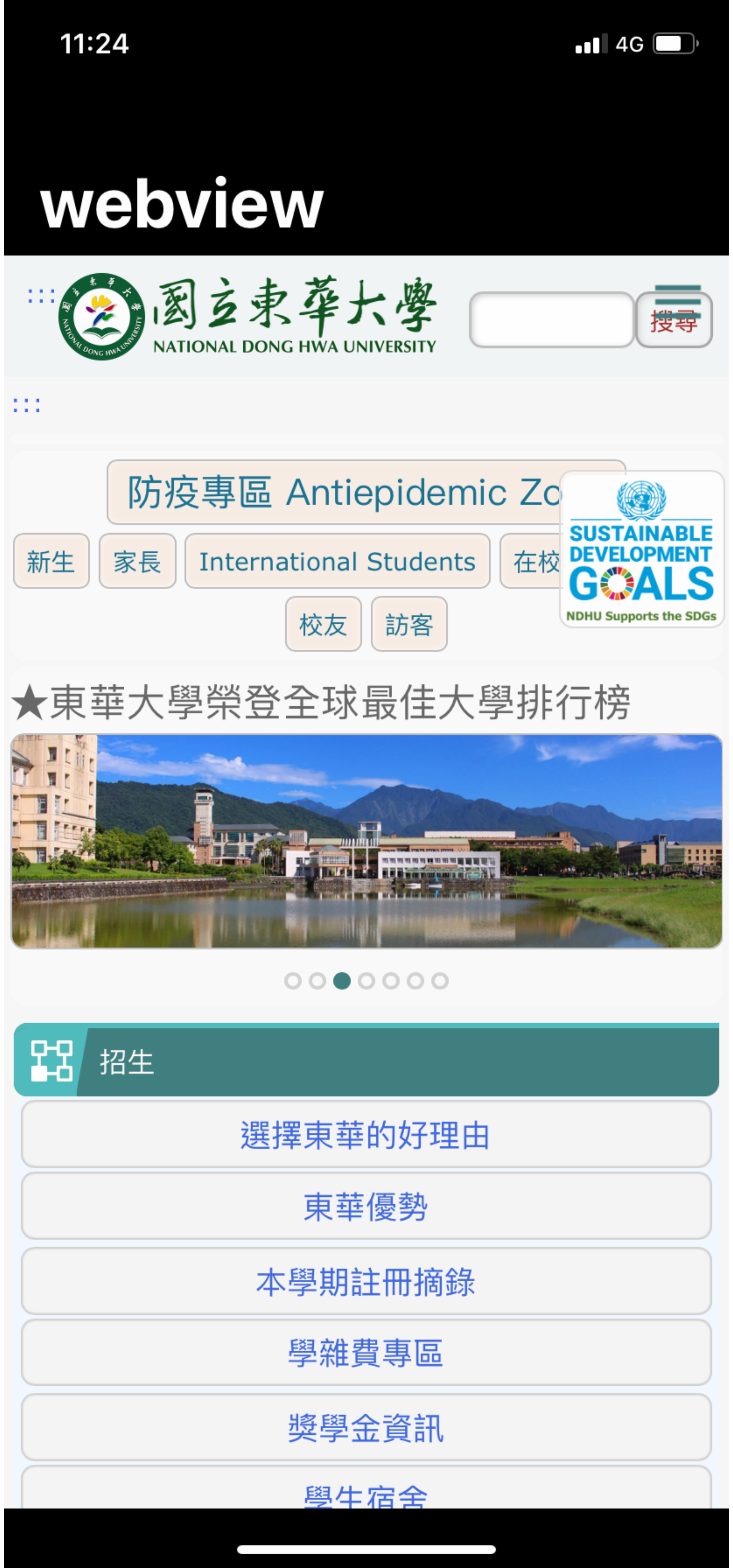


Clustering



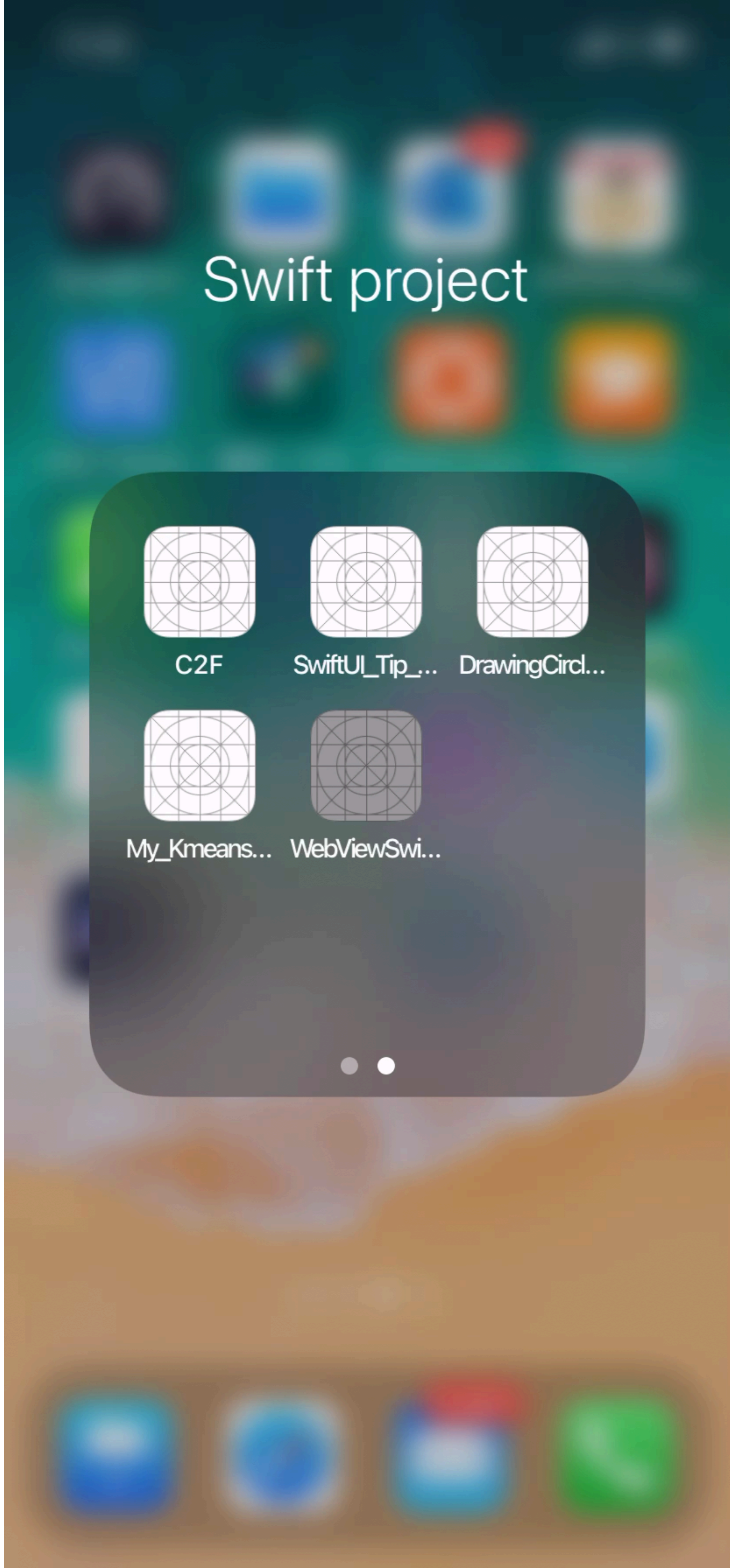
My Web View

- Use swiftUI
- Use Webkit
- Build a WebView App



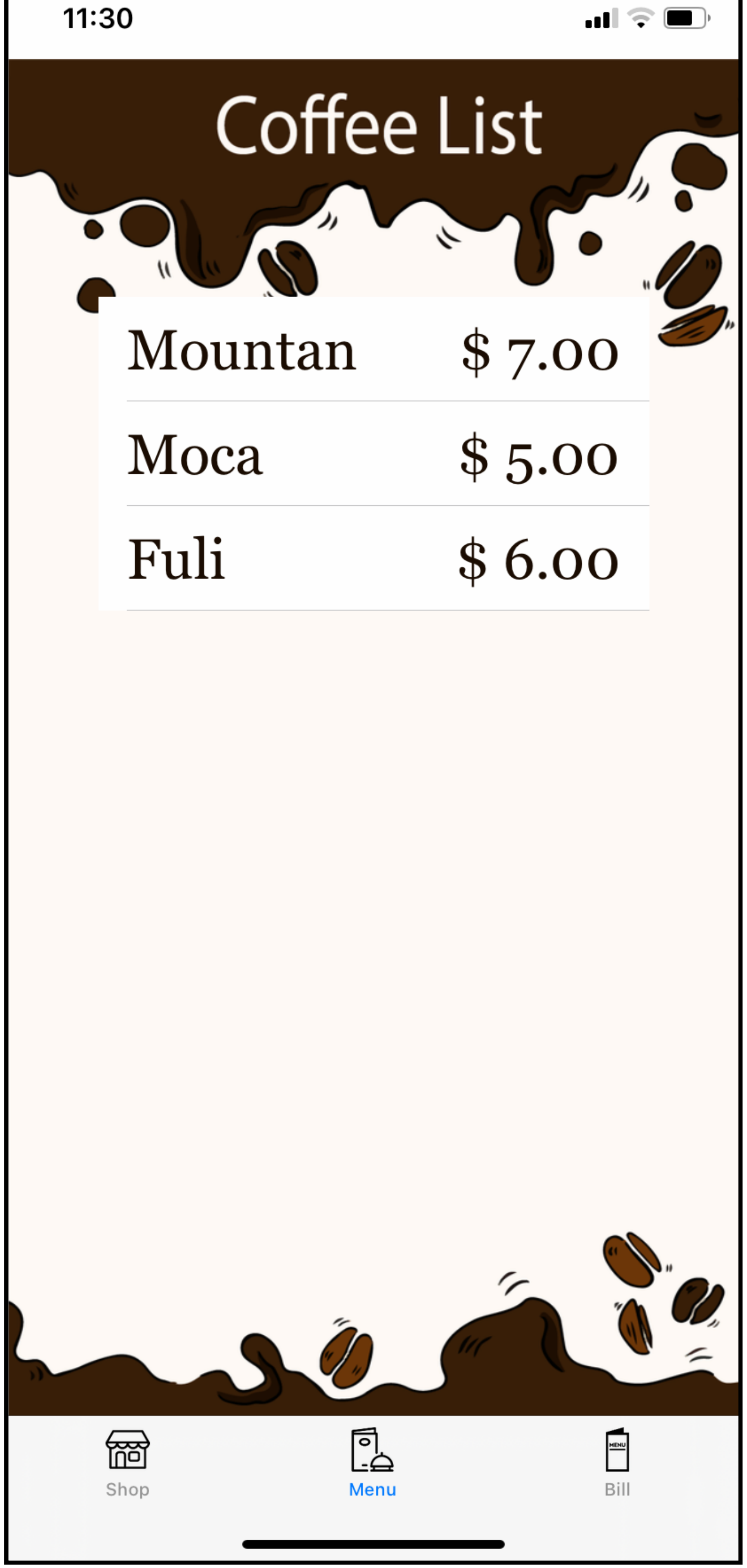
★東華大學榮登全球最佳大學排行榜

My Web View


Swift project
C2F
SwiftUI_Tip_...
DrawingCircl...
My_Kmeans...
WebViewSwi...

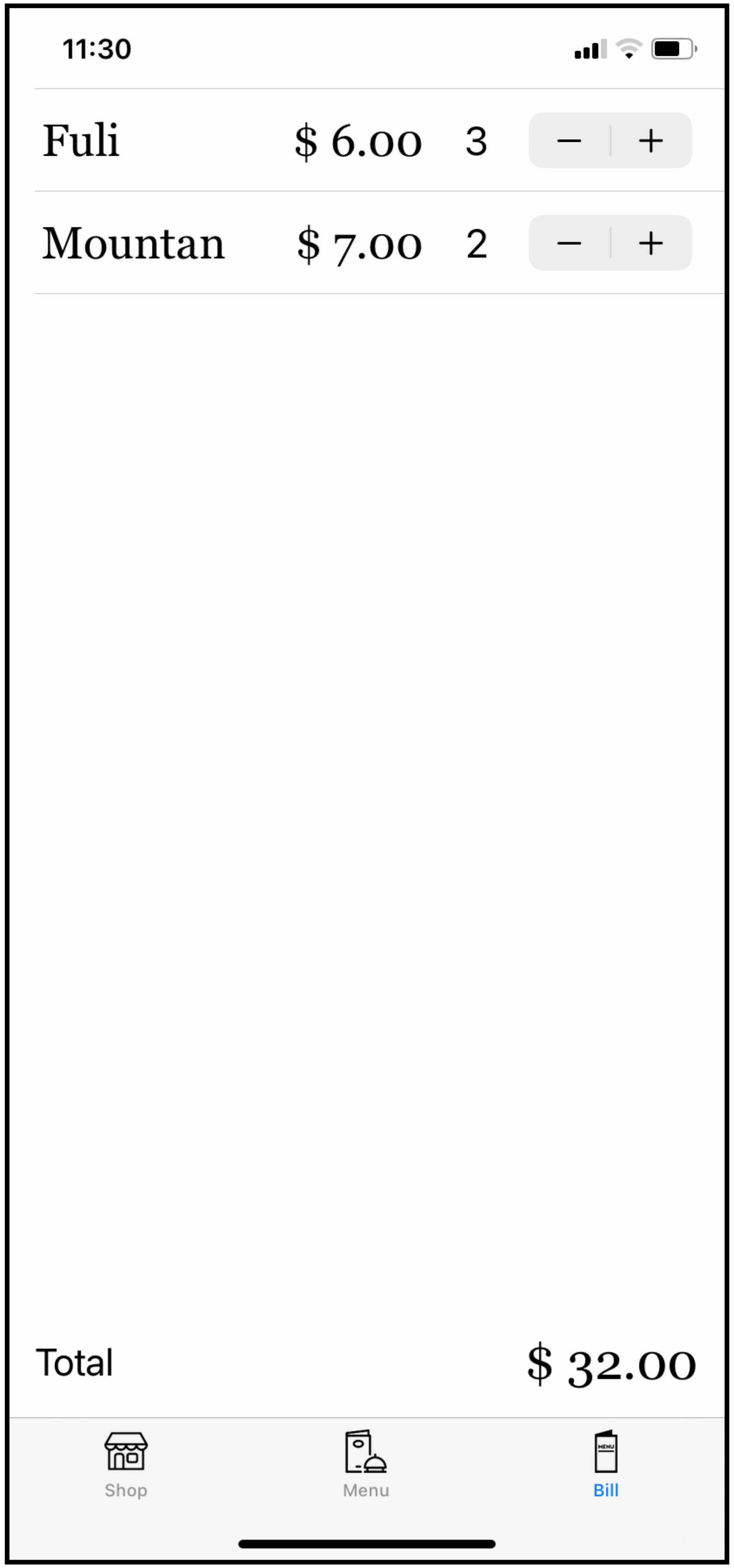
Coffee Shop

- Add products
- Select products
- Increase/ decrease order number
- Bill



Coffee List

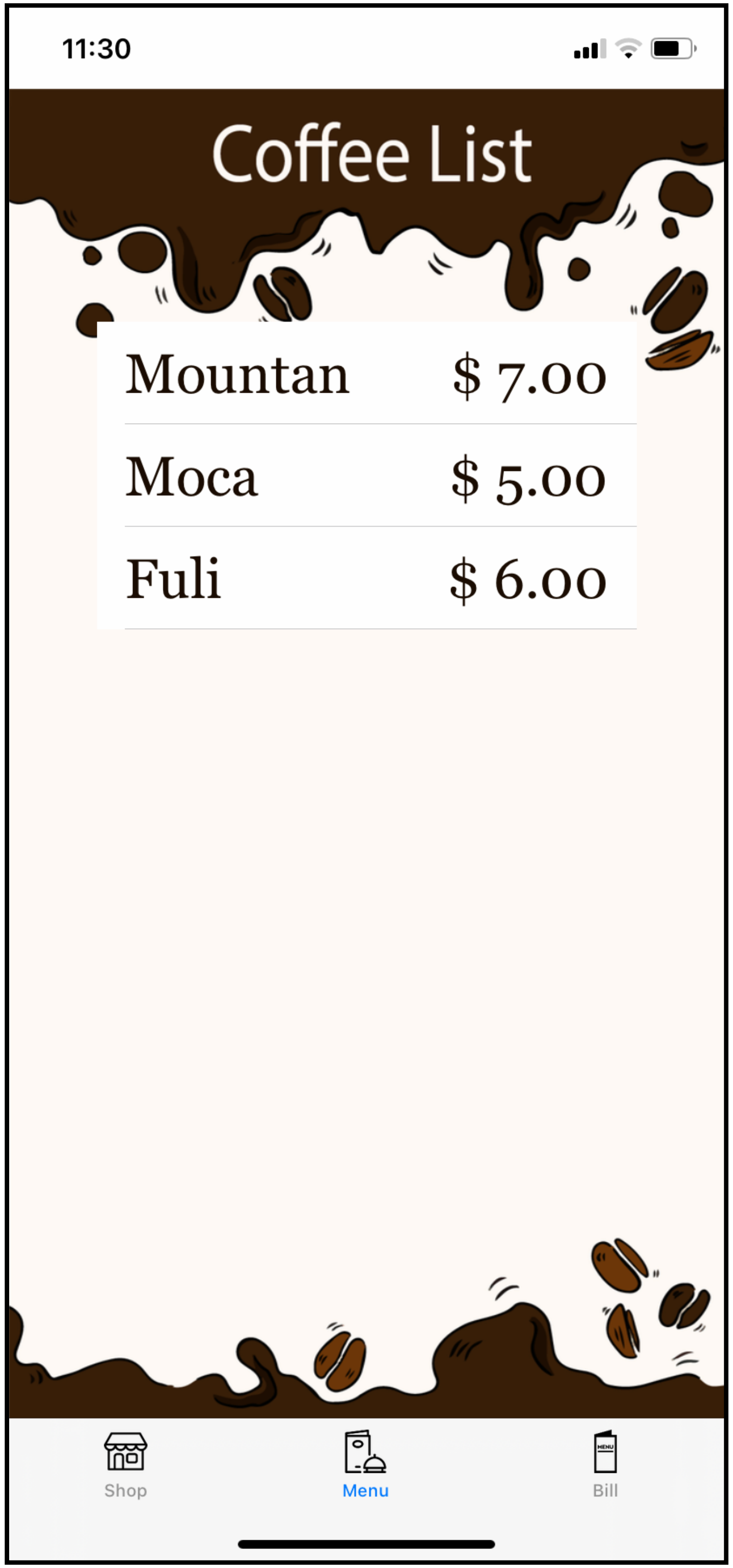
Mountan	$ 7.00
Moca	$ 5.00
Fuli	$ 6.00



11:30

Fuli	$ 6.00	3	− +
Mountan	$ 7.00	2	− +

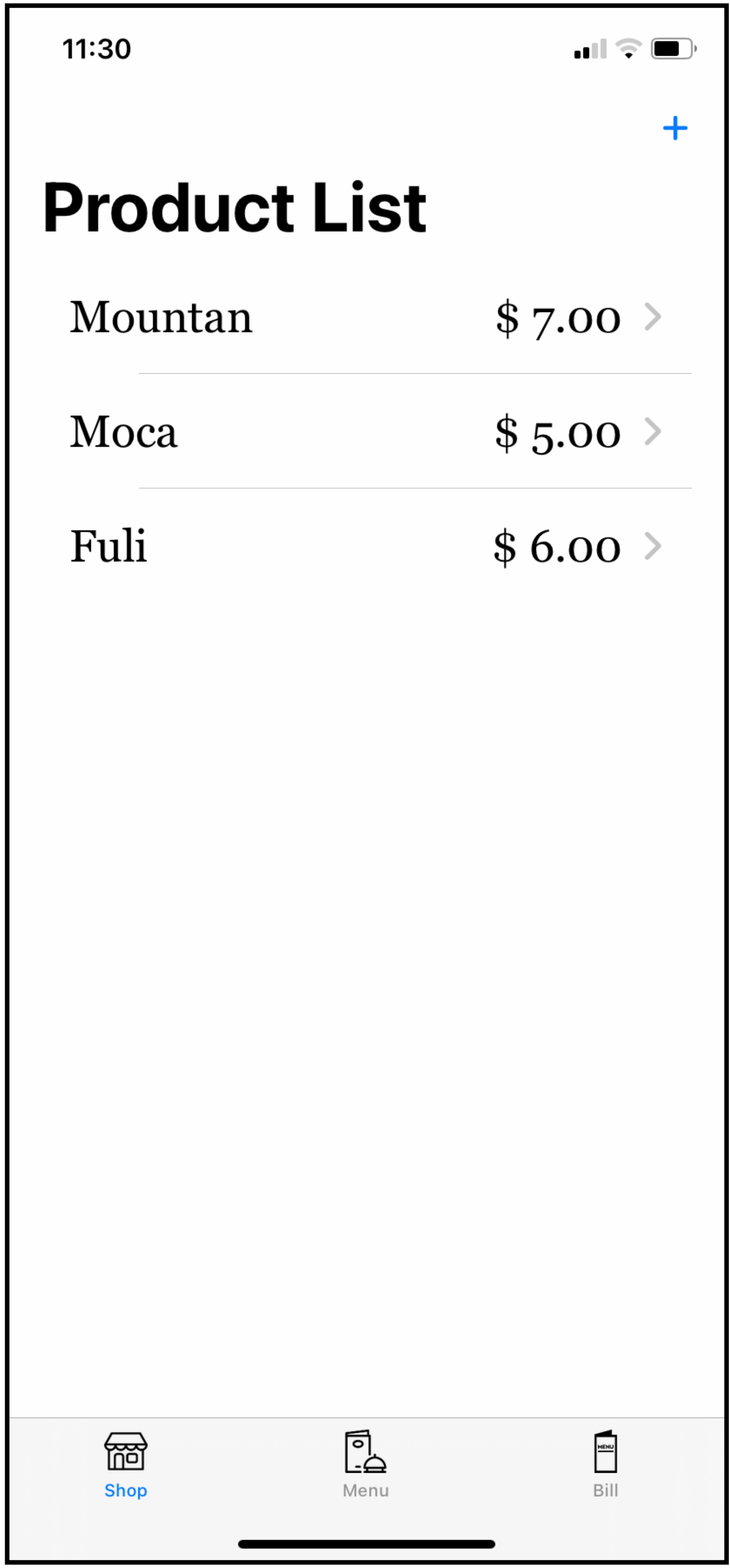
Total $ 32.00



11:30

Coffee List

Mountan	$ 7.00
Moca	$ 5.00
Fuli	$ 6.00



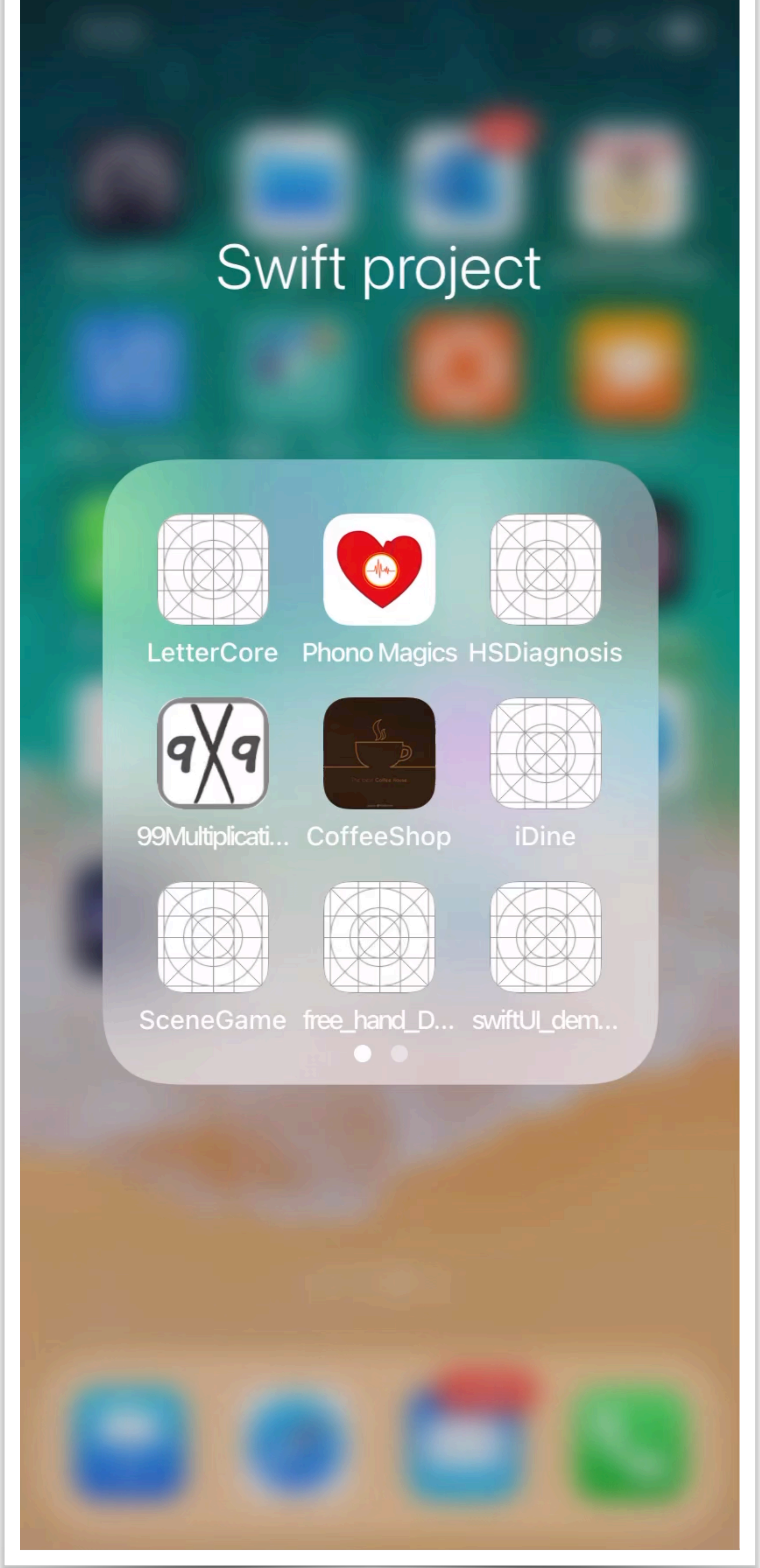
11:30

+

Product List

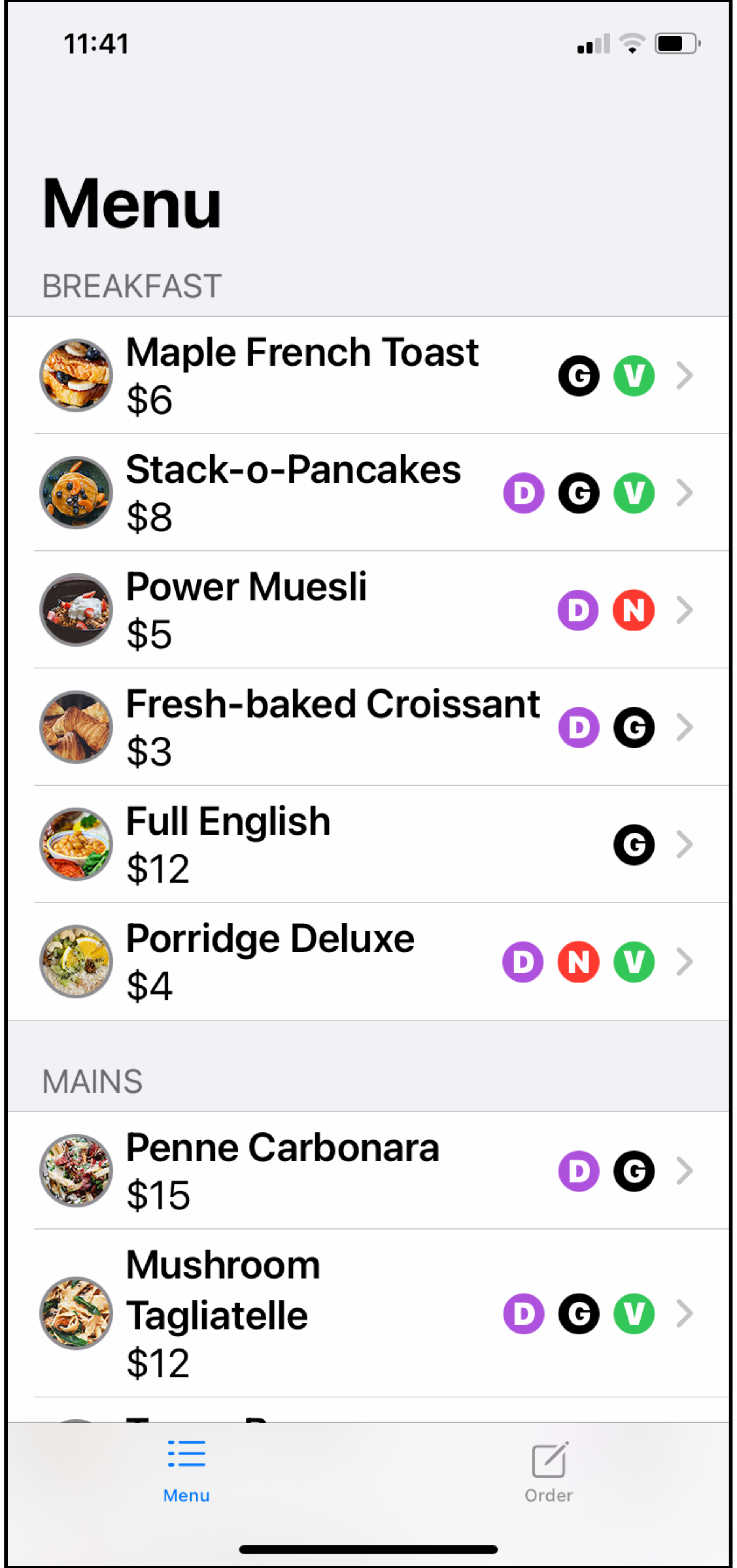
Mountan	$ 7.00 >
Moca	$ 5.00 >
Fuli	$ 6.00 >

Coffee Shop


Swift project
LetterCore
Phono Magics
HSDiagnosis
99Multiplicati...
CoffeeShop
iDine
SceneGame
free_hand_D...
swiftUI_dem...

iDine

- Menu
- Product introduction
- Order
- Edit order
- Summary
- Tip



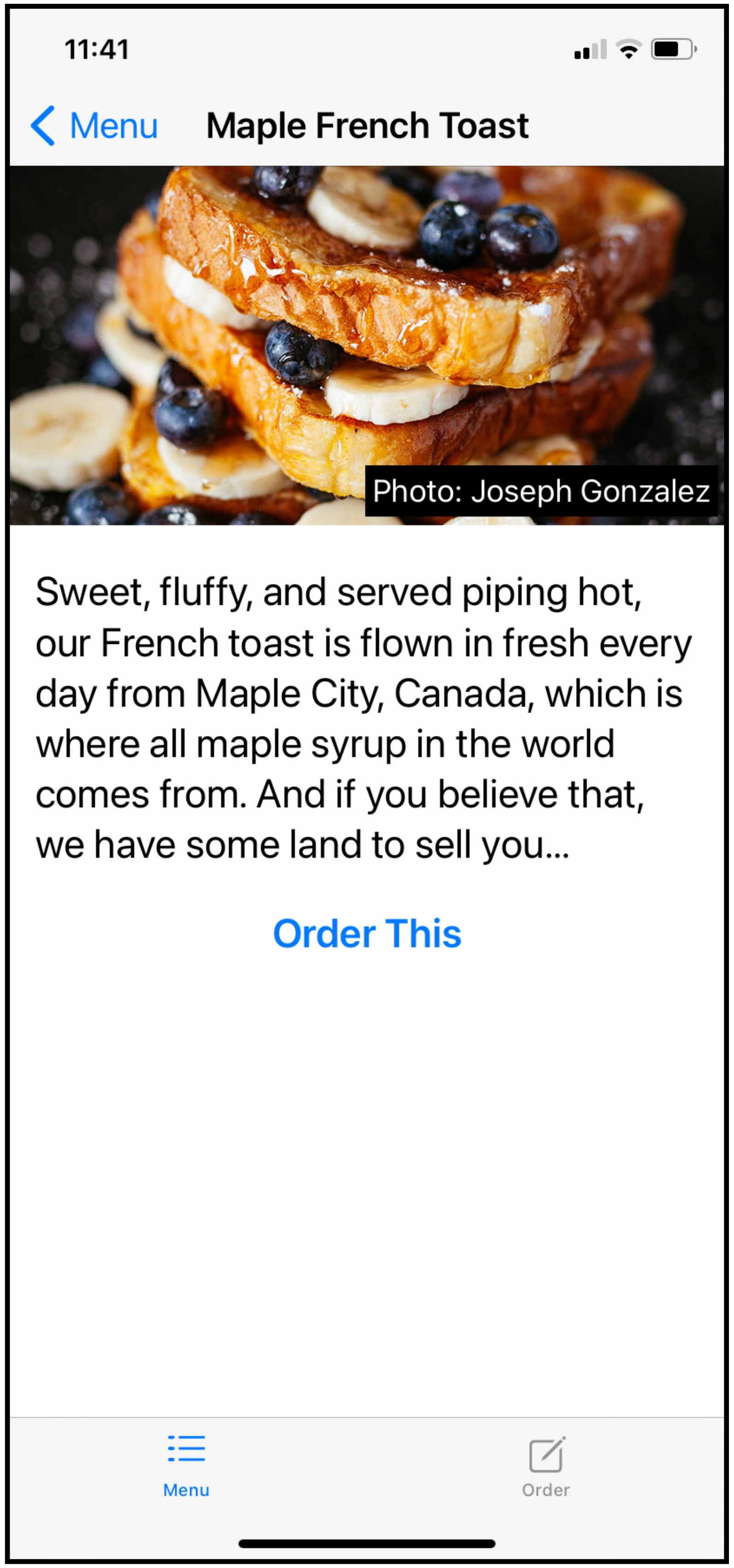
Menu

BREAKFAST

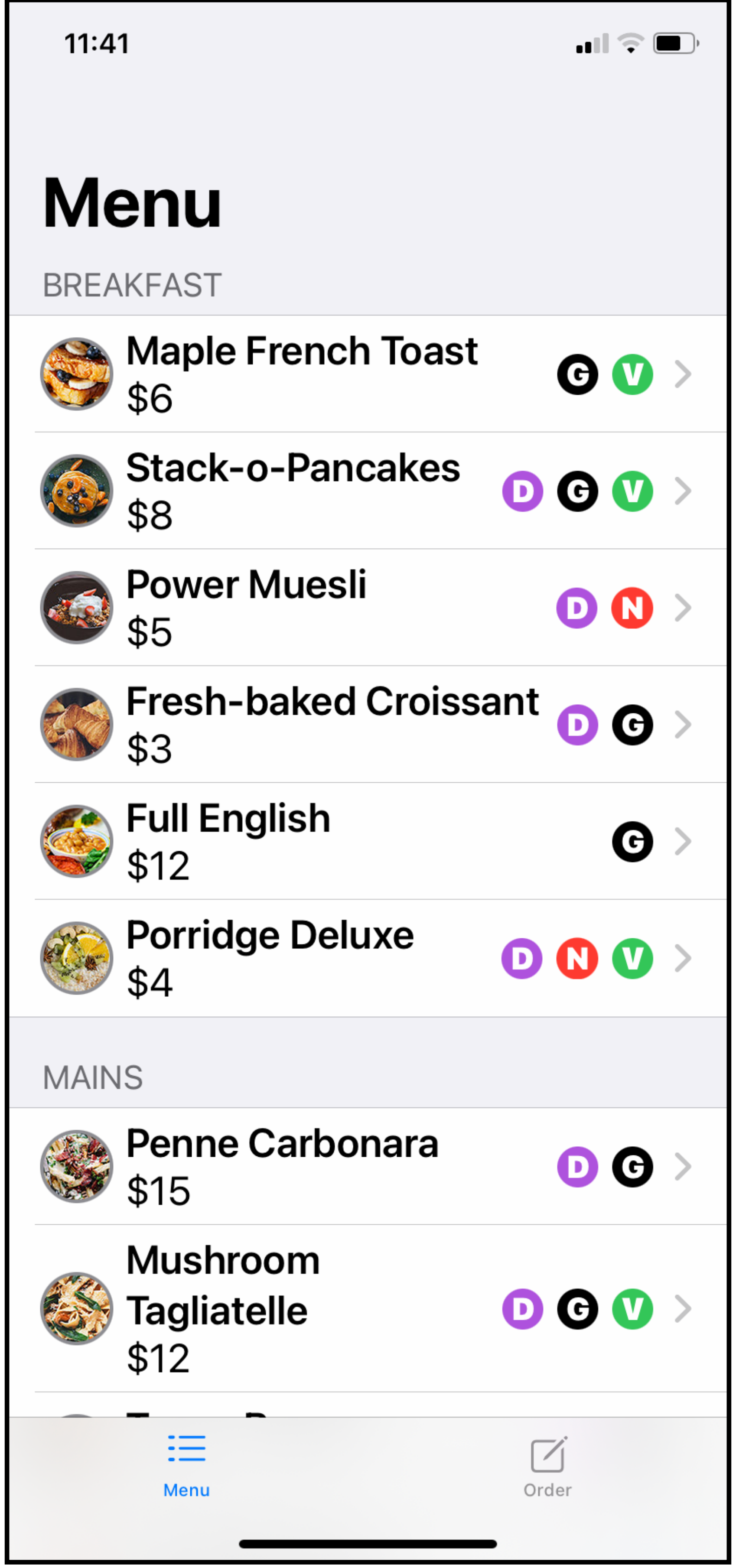
Maple French Toast	$6	G V
Stack-o-Pancakes	$8	D G V
Power Muesli	$5	D N
Fresh-baked Croissant	$3	D G
Full English	$12	G
Porridge Deluxe	$4	D N V

MAINS

Penne Carbonara	$15	D G
Mushroom Tagliatelle	$12	D G V



Sweet, fluffy, and served piping hot, our French toast is flown in fresh every day from Maple City, Canada, which is where all maple syrup in the world comes from. And if you believe that, we have some land to sell you...

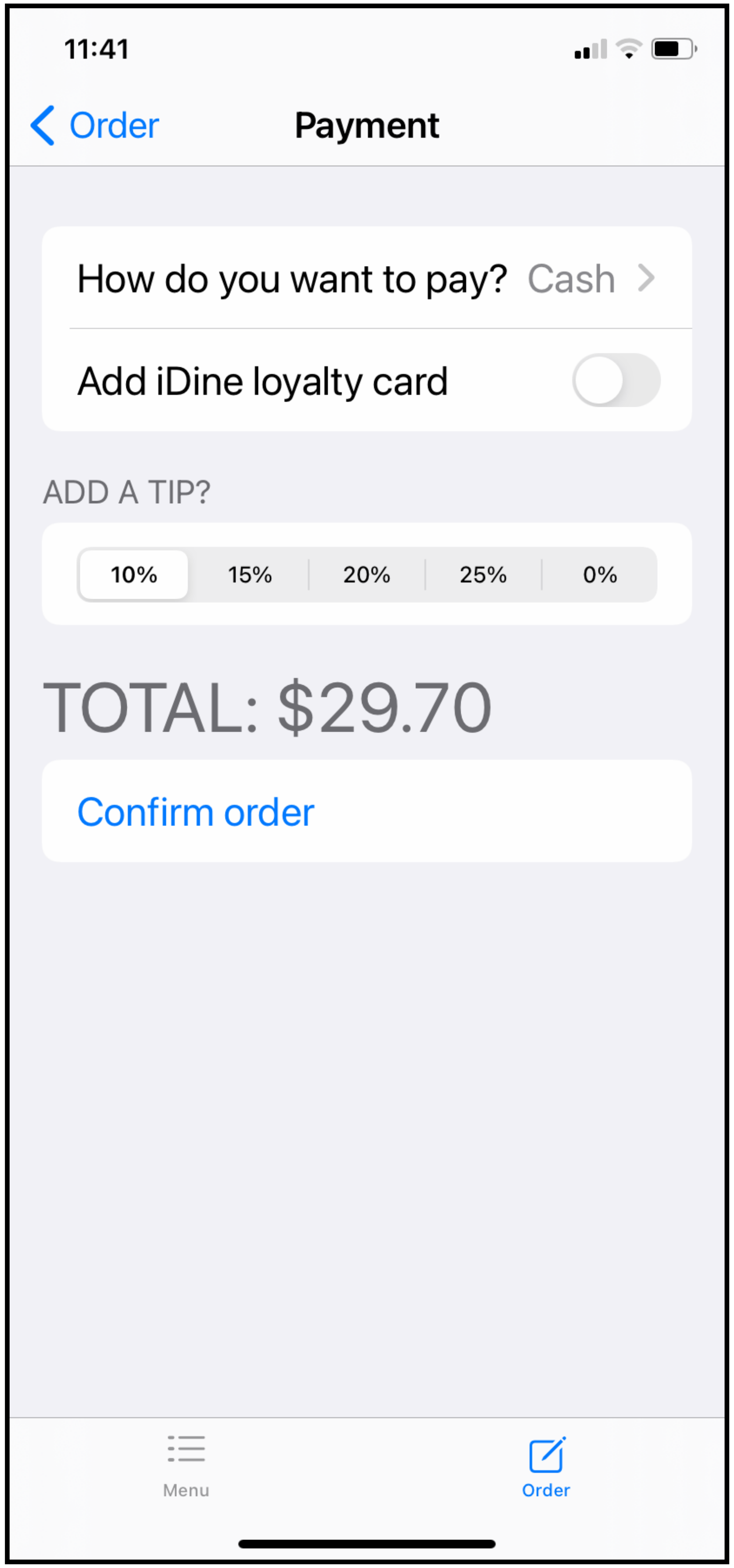


BREAKFAST

Maple French Toast	$6	G V
Stack-o-Pancakes	$8	D G V
Power Muesli	$5	D N
Fresh-baked Croissant	$3	D G
Full English	$12	G
Porridge Deluxe	$4	D N V

MAINS

Penne Carbonara	$15	D G
Mushroom Tagliatelle	$12	D G V



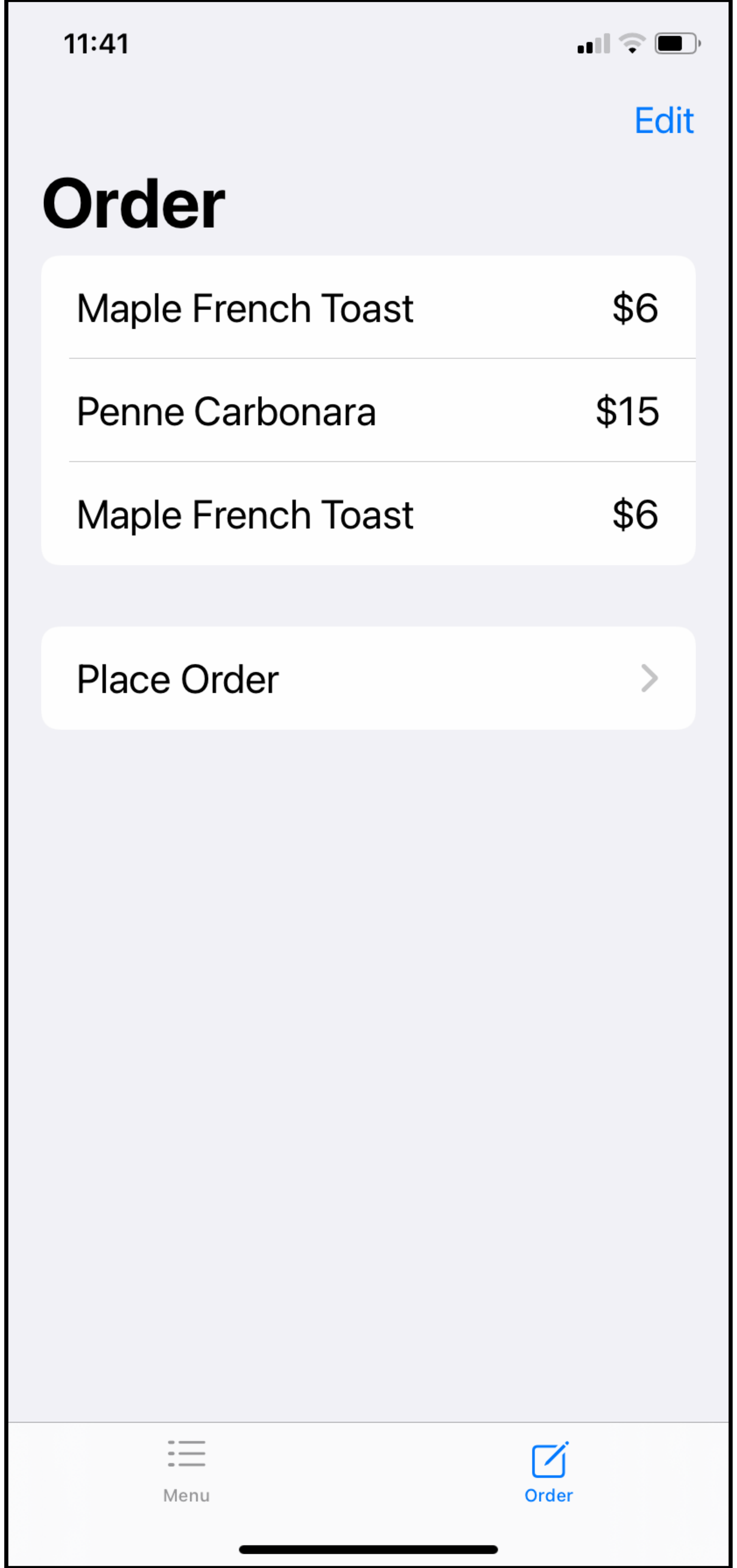
How do you want to pay? Cash

Add iDine loyalty card ○

ADD A TIP?

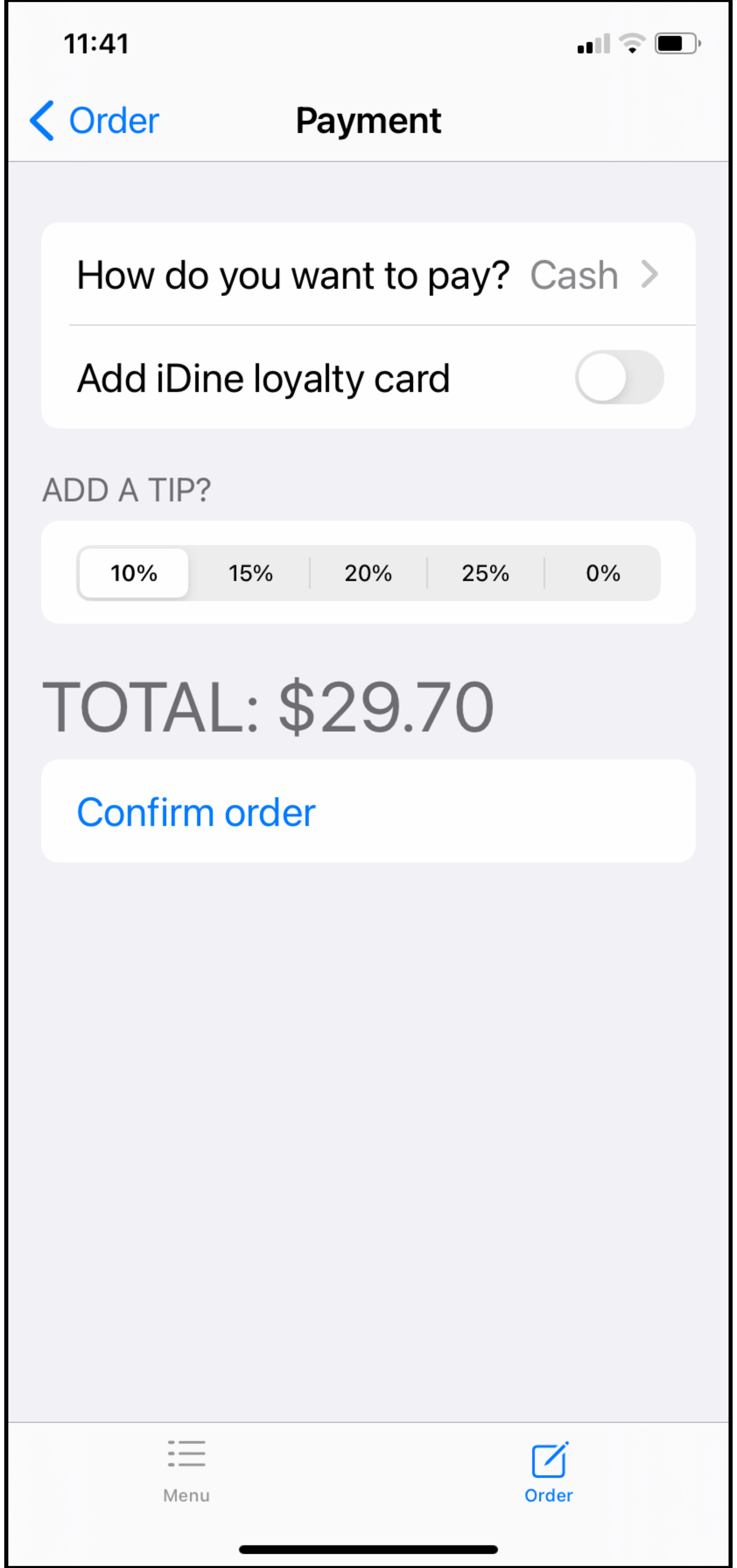
[✓ 10%] 15% 20% 25% 0%

TOTAL: $29.70



Order

Maple French Toast	$6
Penne Carbonara	$15
Maple French Toast	$6



Payment

How do you want to pay? Cash

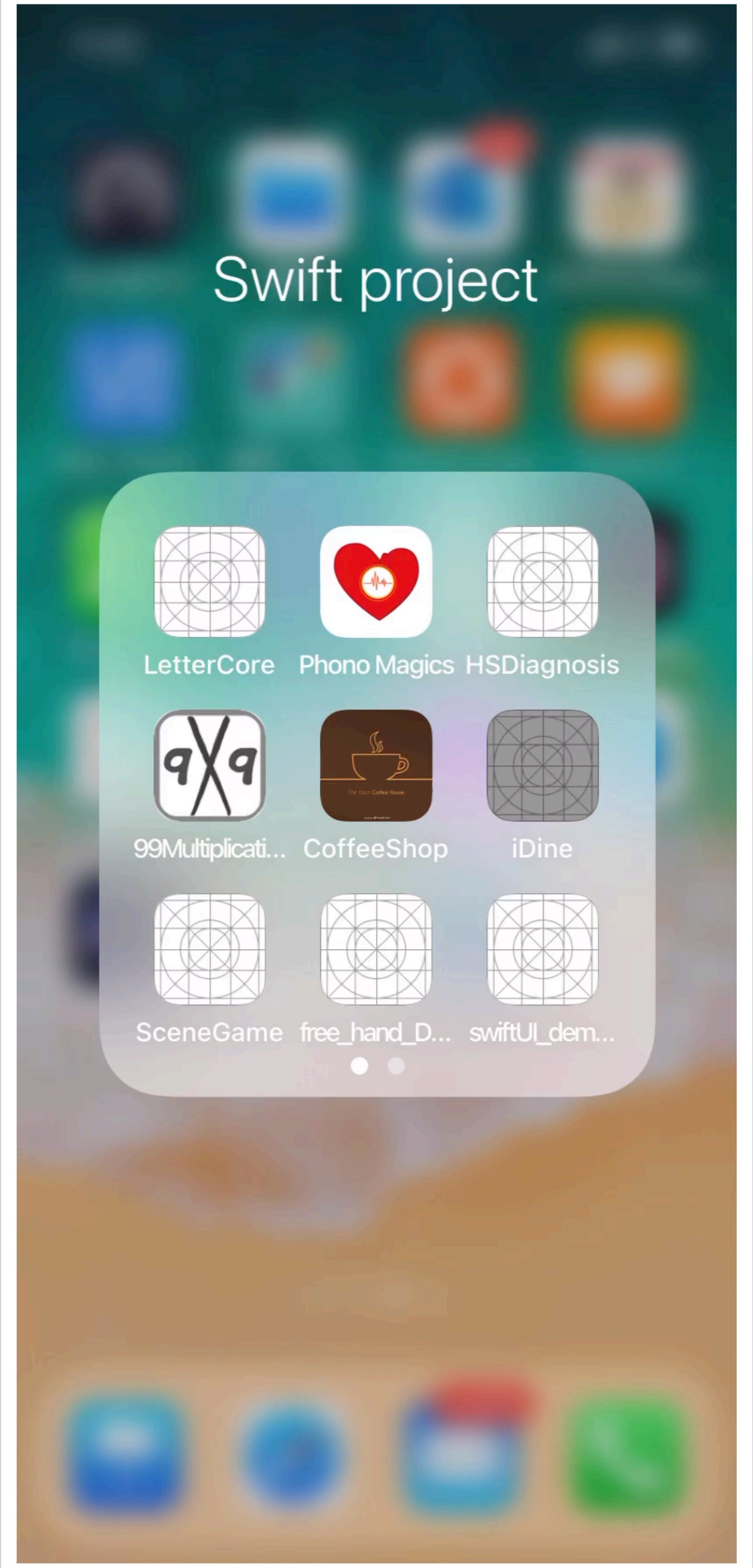
Add iDine loyalty card ○

ADD A TIP?

[✓ 10%] 15% 20% 25% 0%

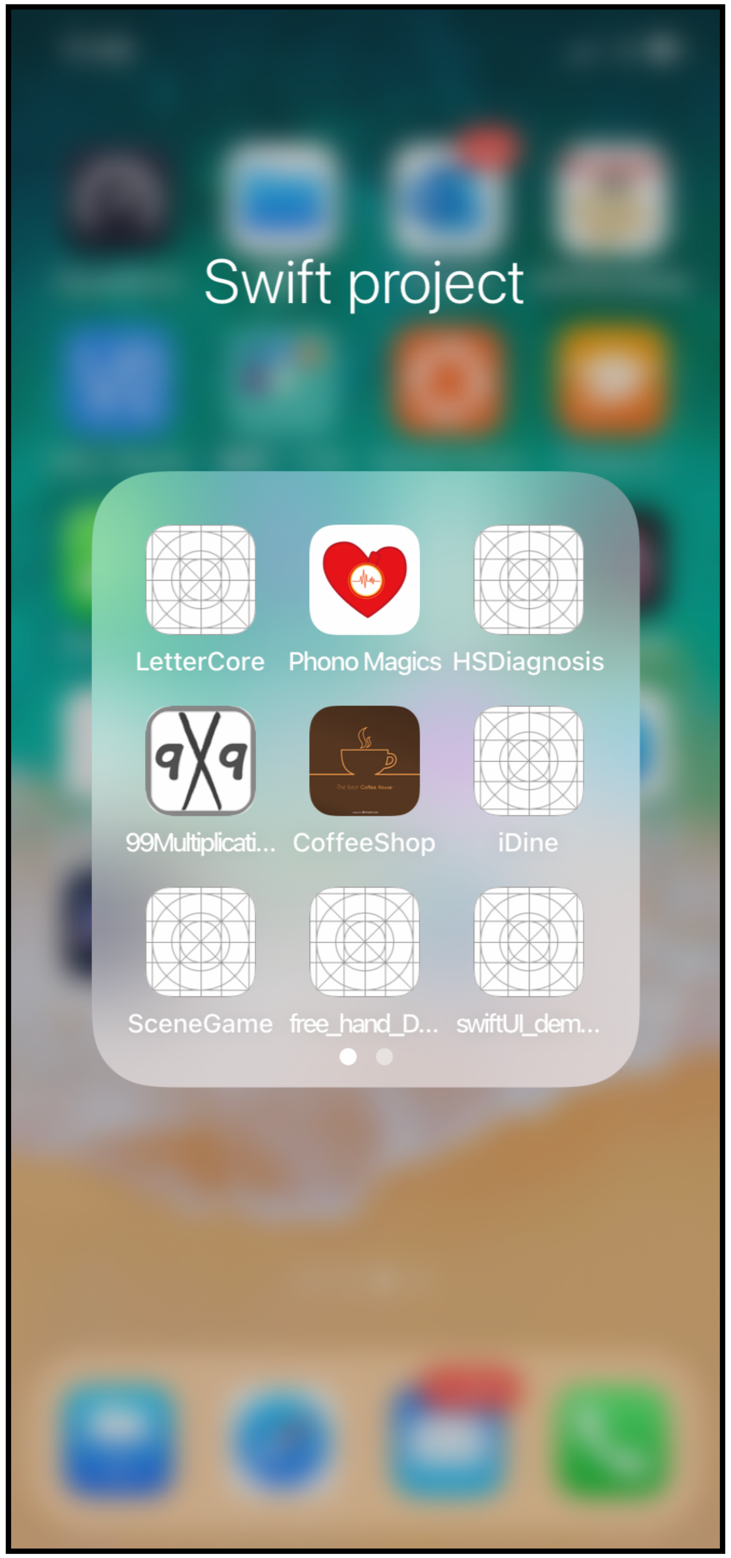
TOTAL: $29.70

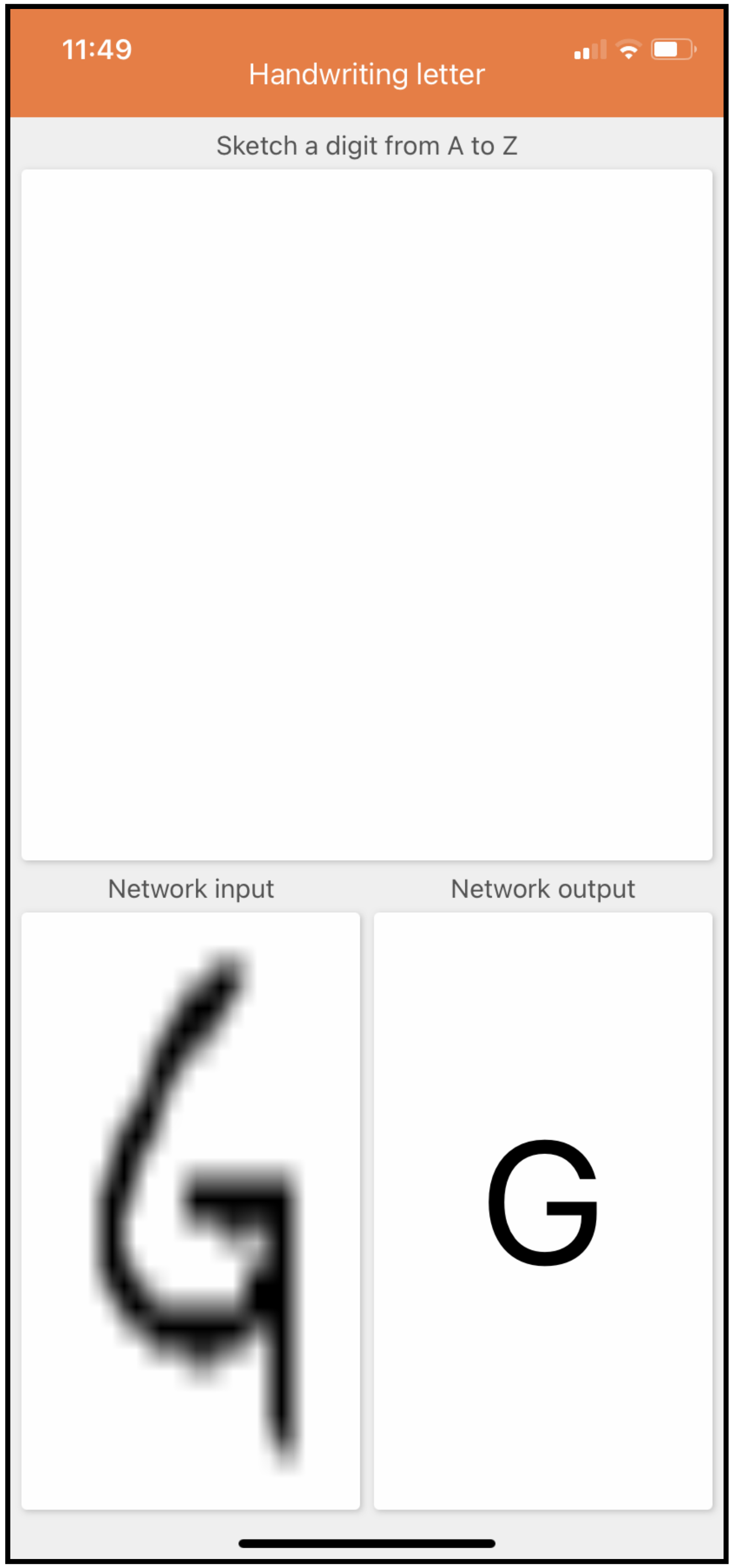
Breakfast order



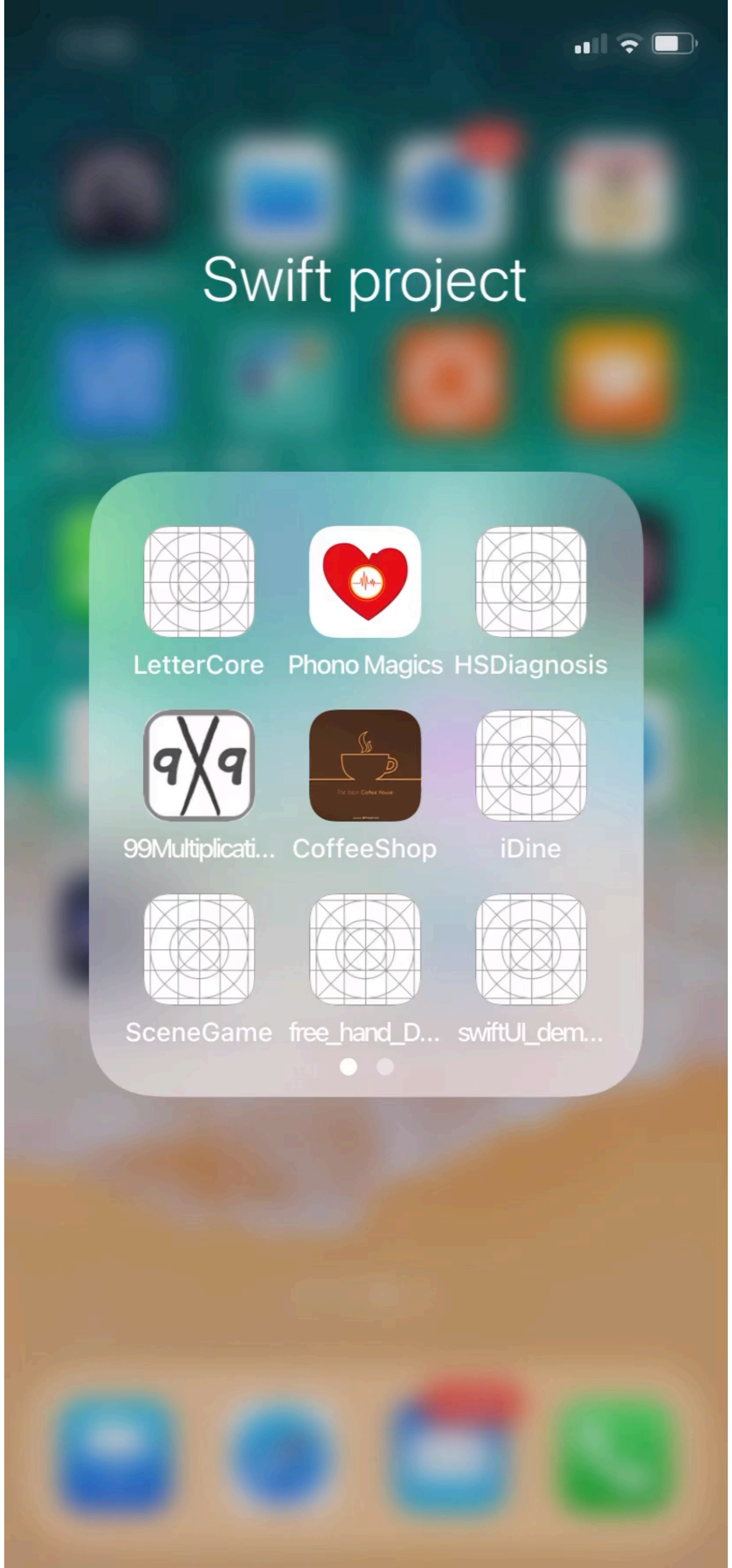
Hand Writing Character Recognition

- Hand writing
- Deep learning
- Deep CNN
- AI Pattern Recognition



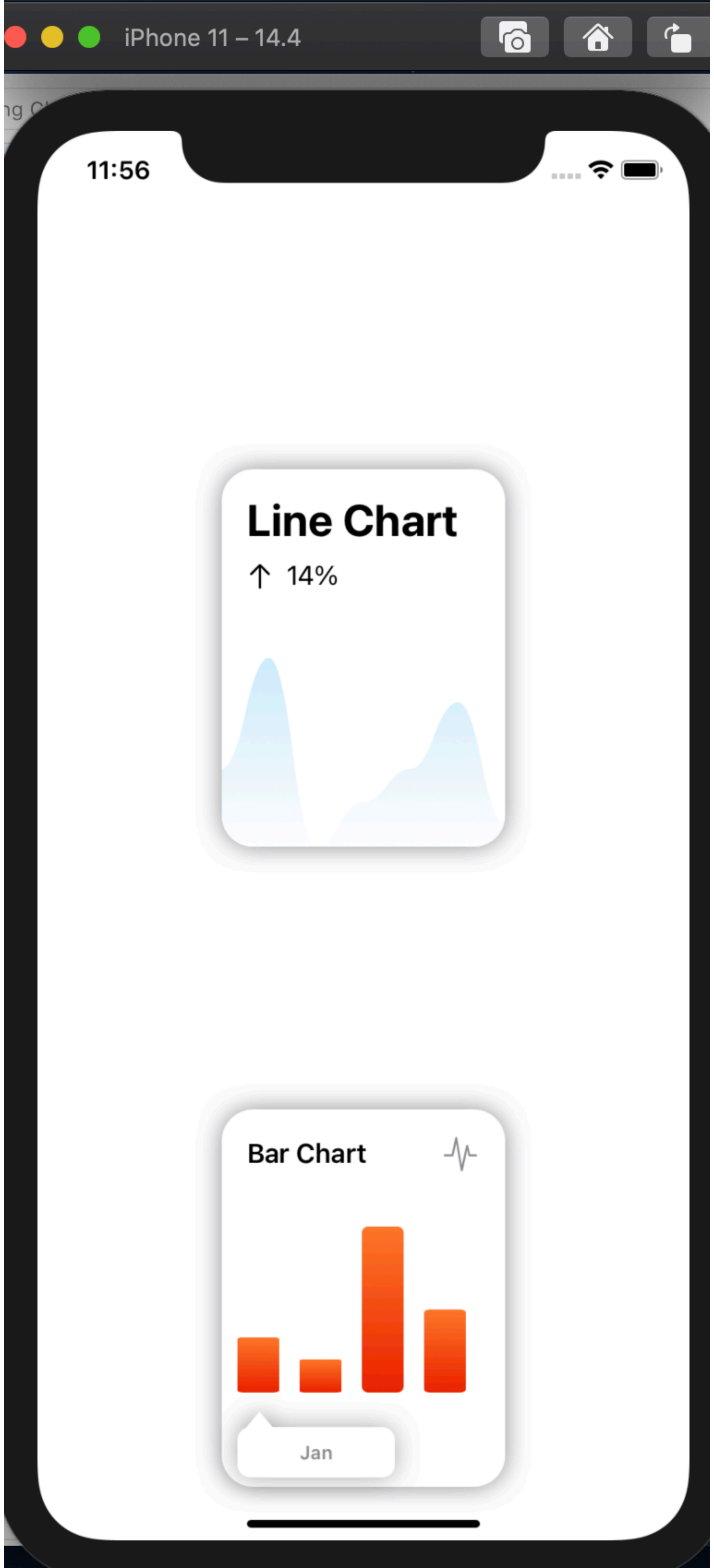


Hand Writing Character Recognition

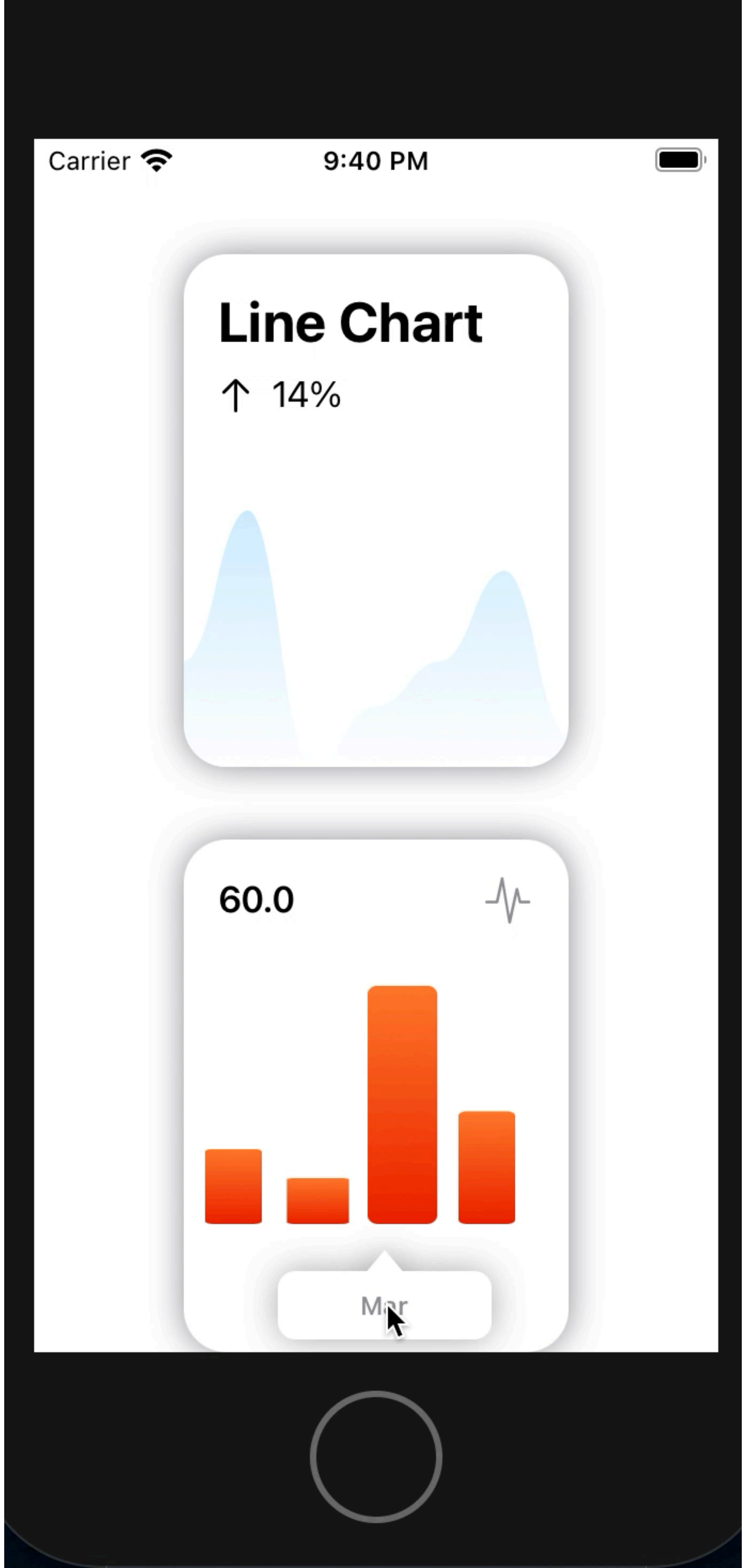


Line Chart and Bar

- Display line for connecting points
- Display Bar Chart
- Use dependency

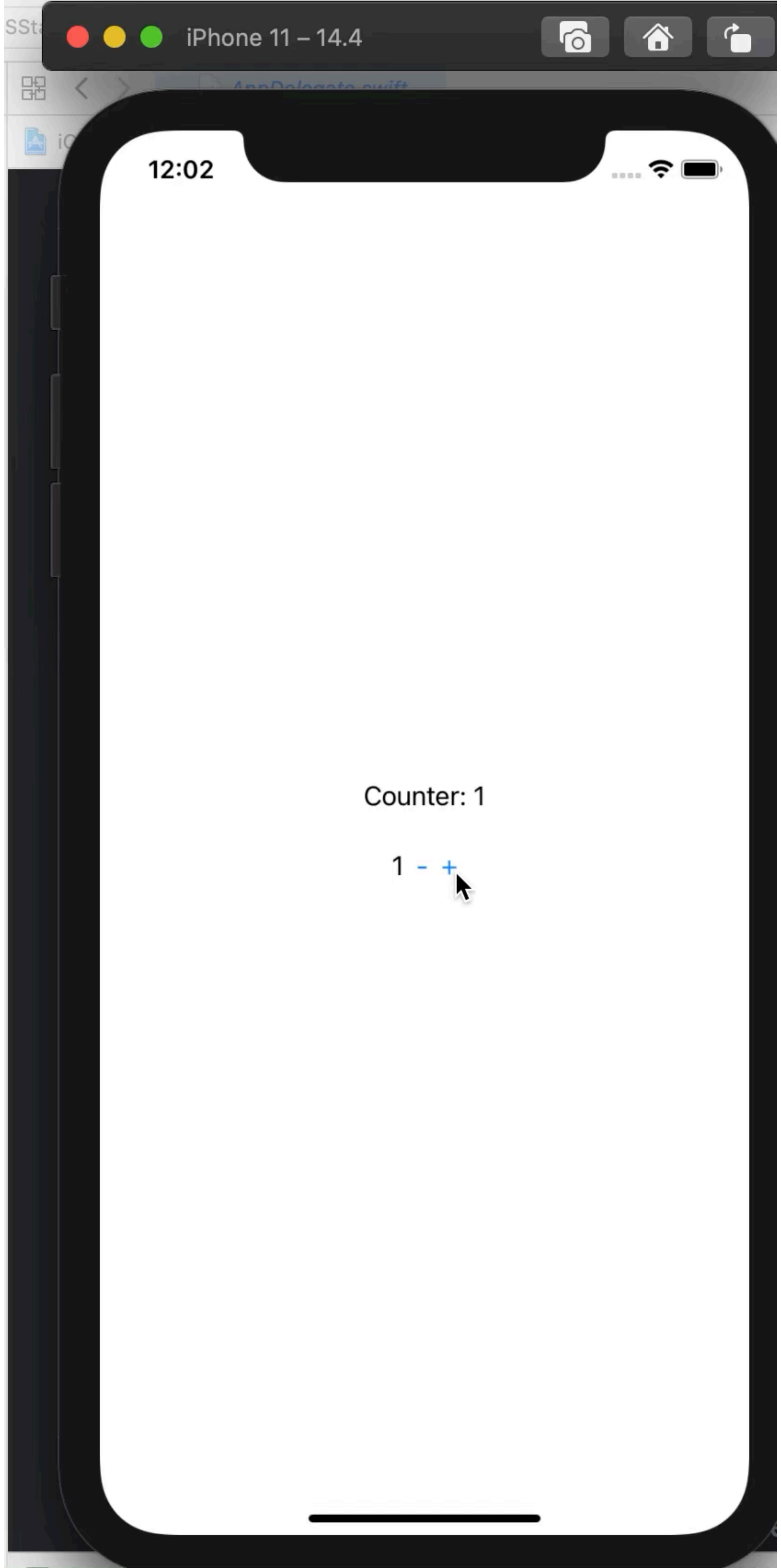


Line Chart and Bar

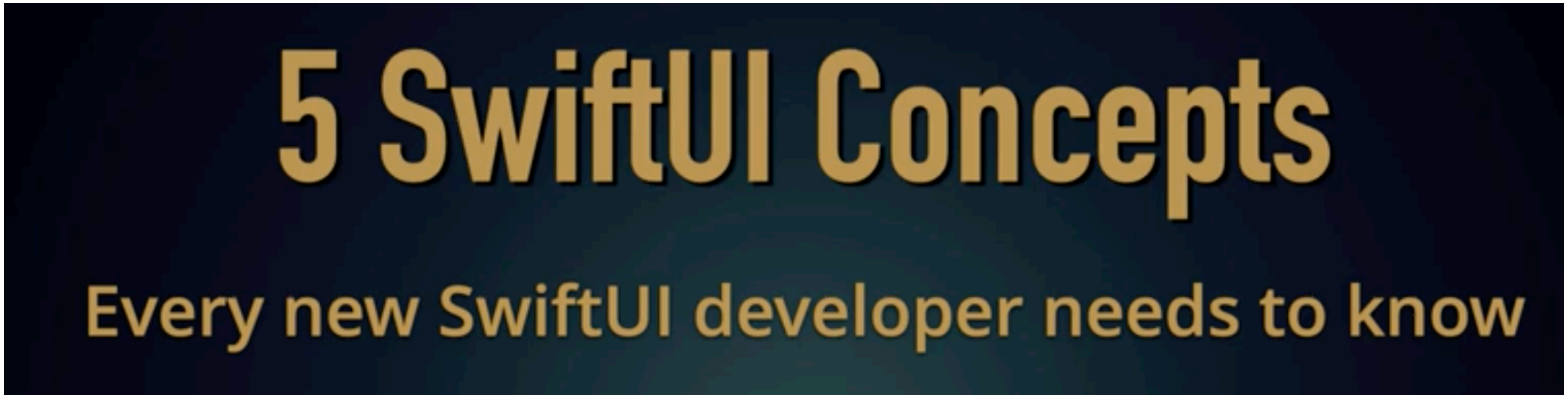


iOS state binding example

- Demo statebinding

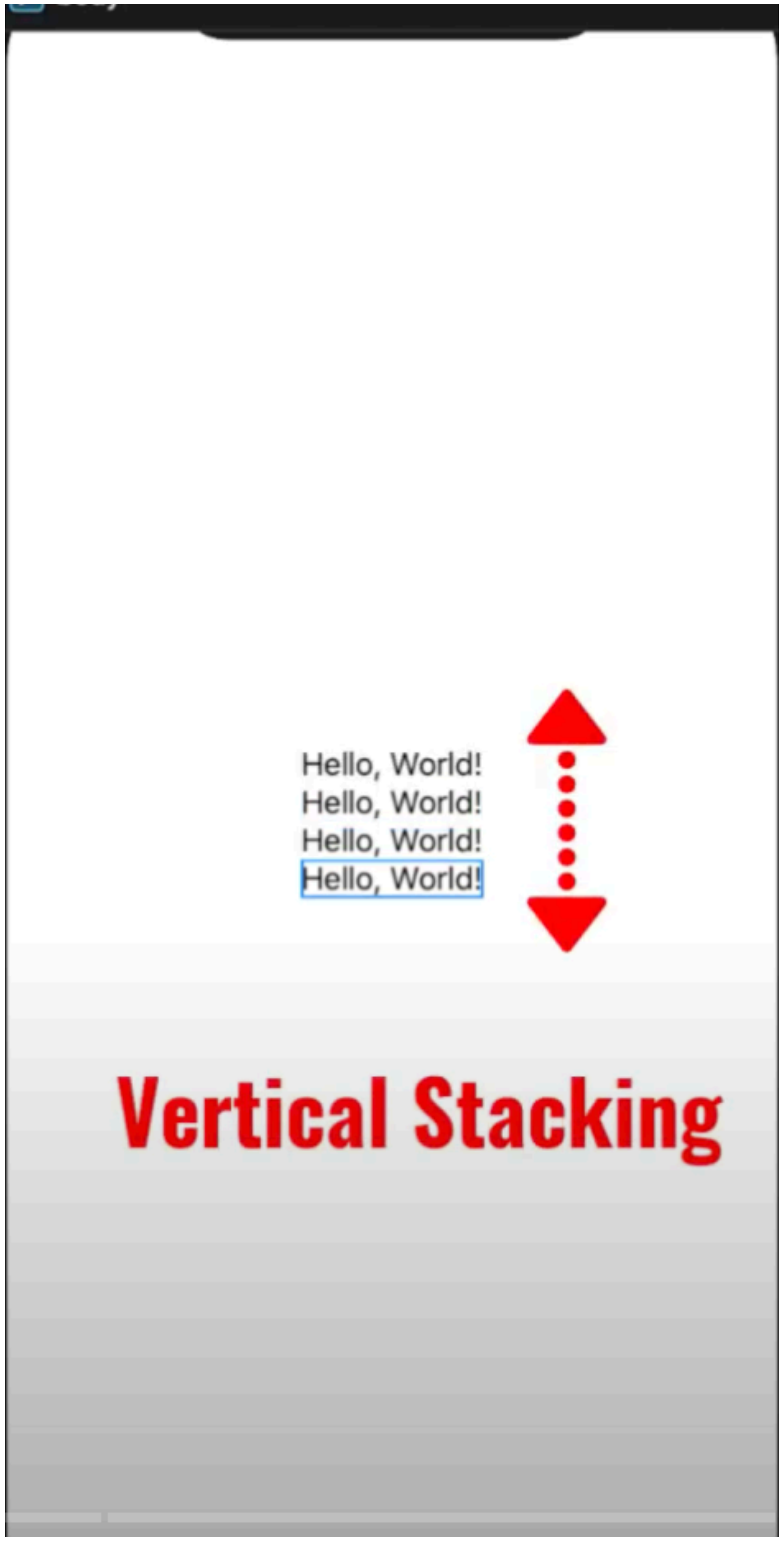


https://www.youtube.com/watch?v=51xIHDm_BDs



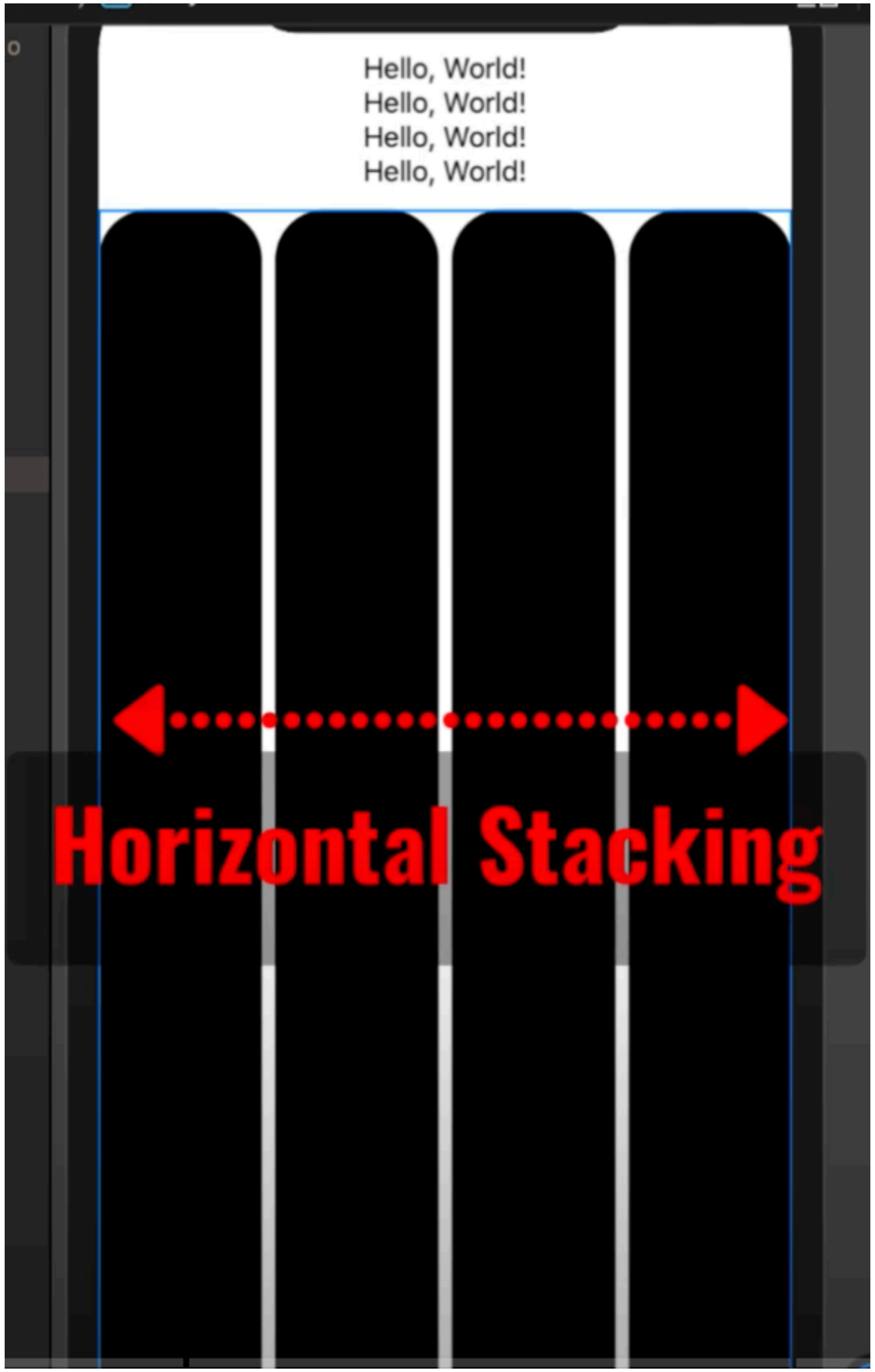
5 SwiftUI Concepts

Every new SwiftUI developer needs to know



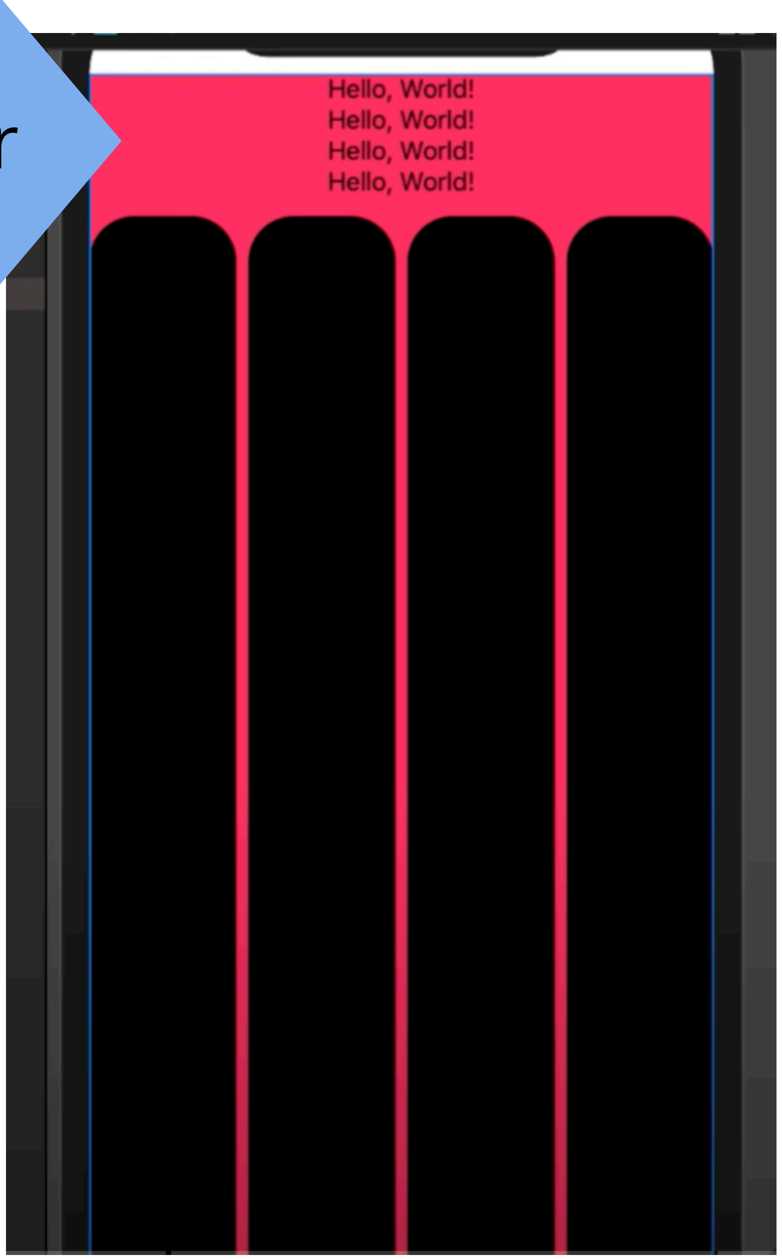
```
Hello, World!
Hello, World!
Hello, World!
Hello, World!
```

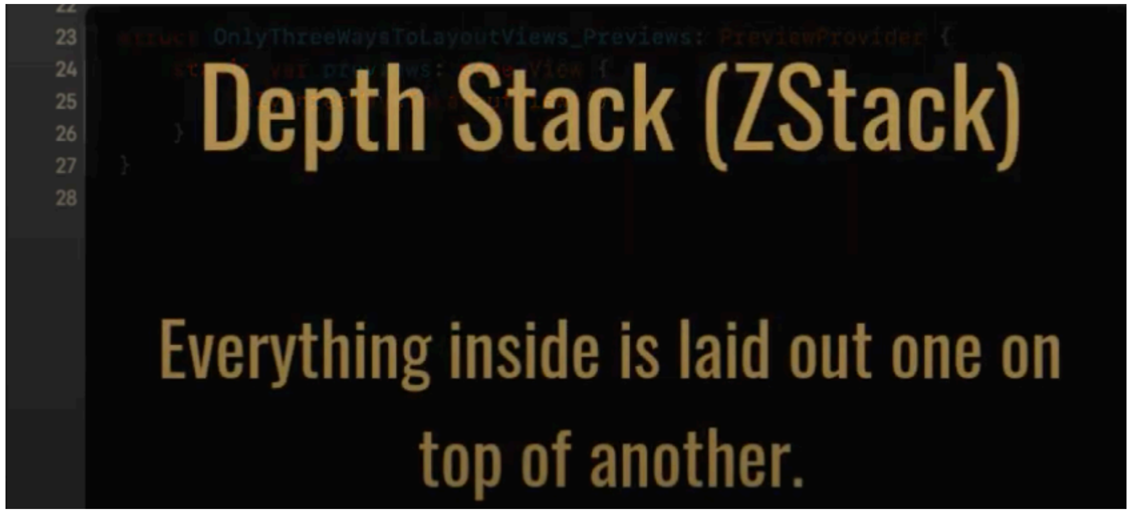
Vertical Stacking



Hello, World!
Hello, World!
Hello, World!
Hello, World!

Bottom Layer


Hello, World!
Hello, World!
Hello, World!
Hello, World!

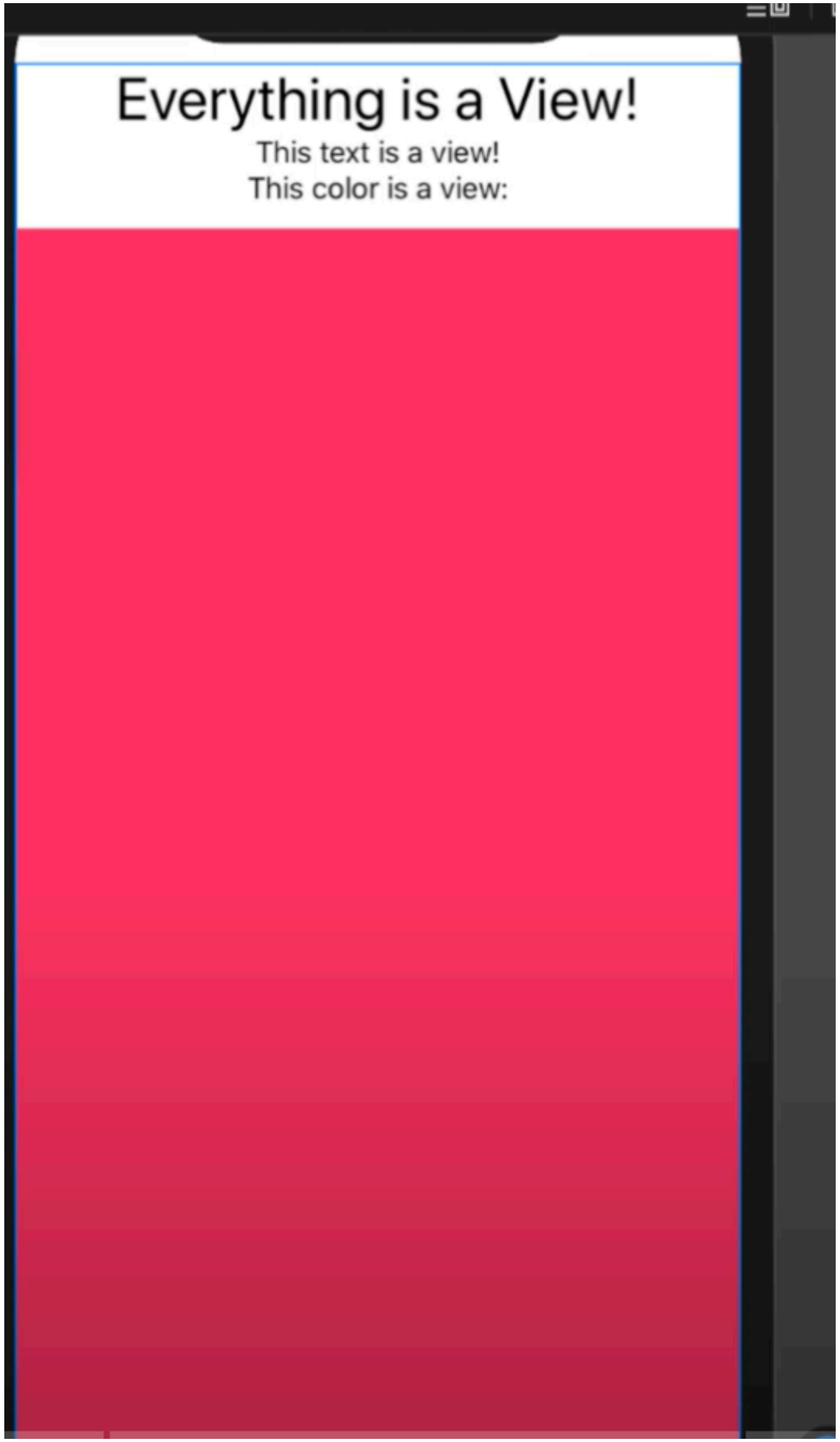

Depth Stack (ZStack)

Everything inside is laid out one on top of another.

CONCEPT

2

Everything is a View



Everything is a View!

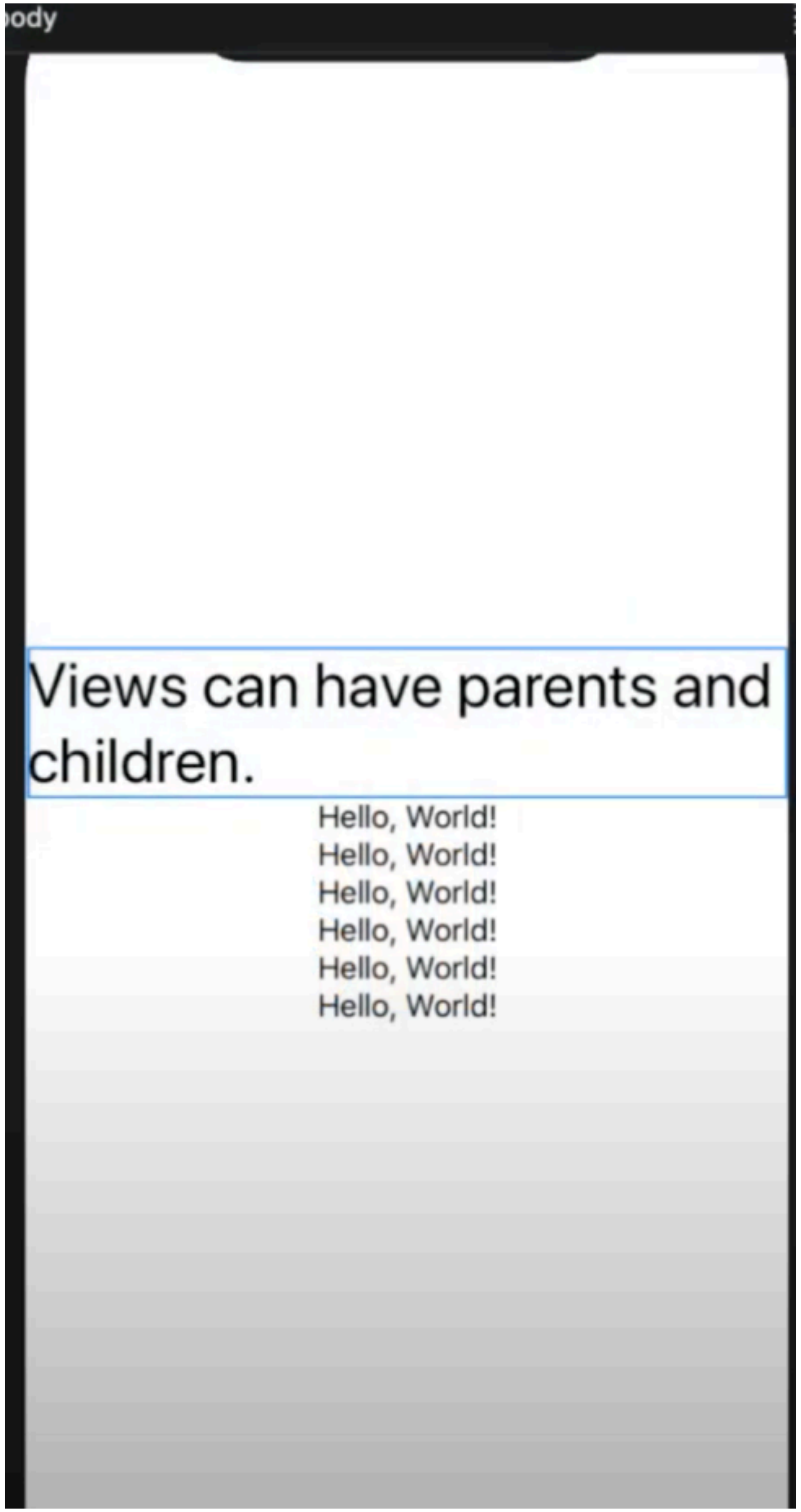
This text is a view!

This color is a view:

CONCEPT

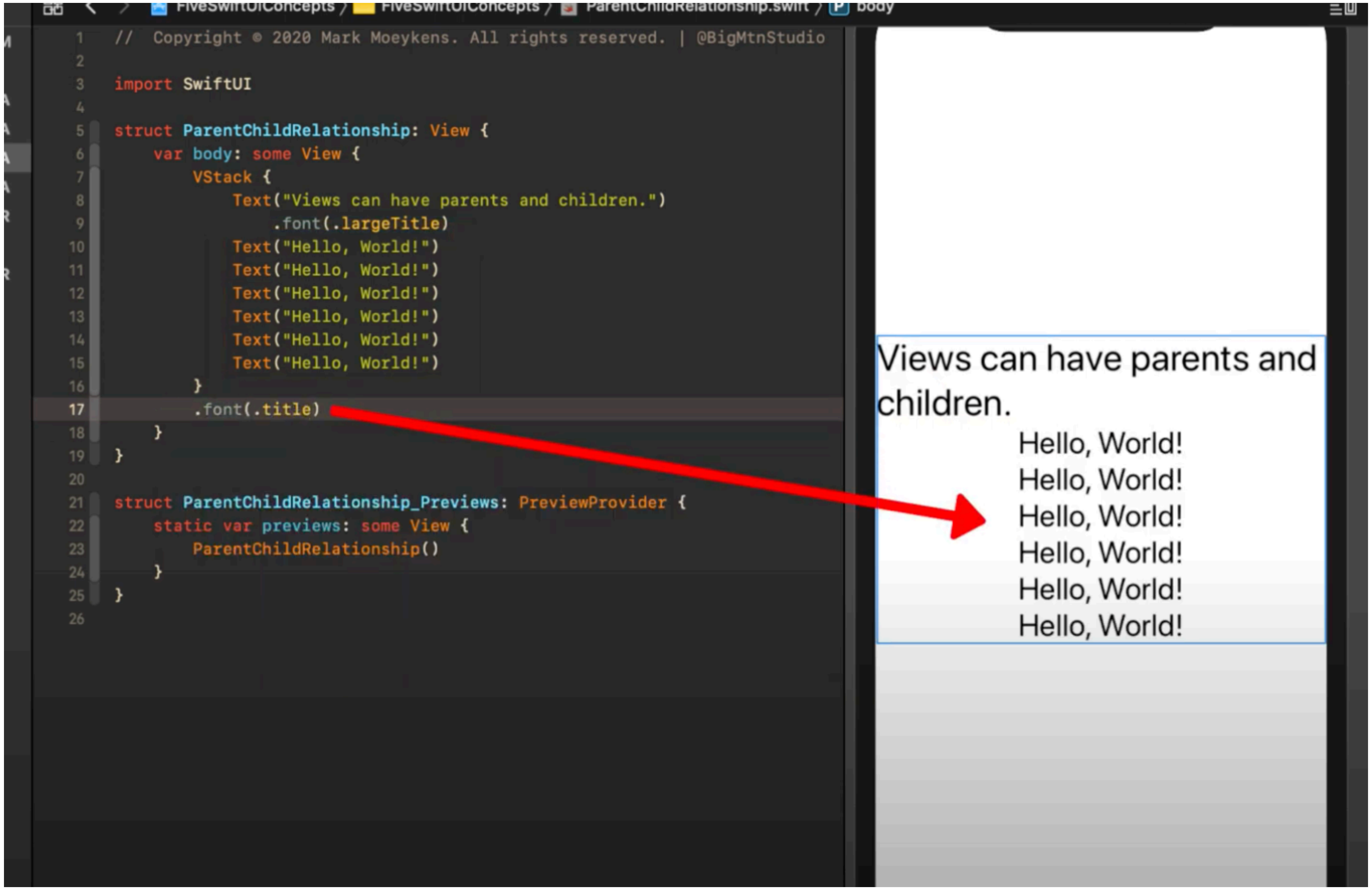
3

Parent – Child Relationships



Views can have parents and children.

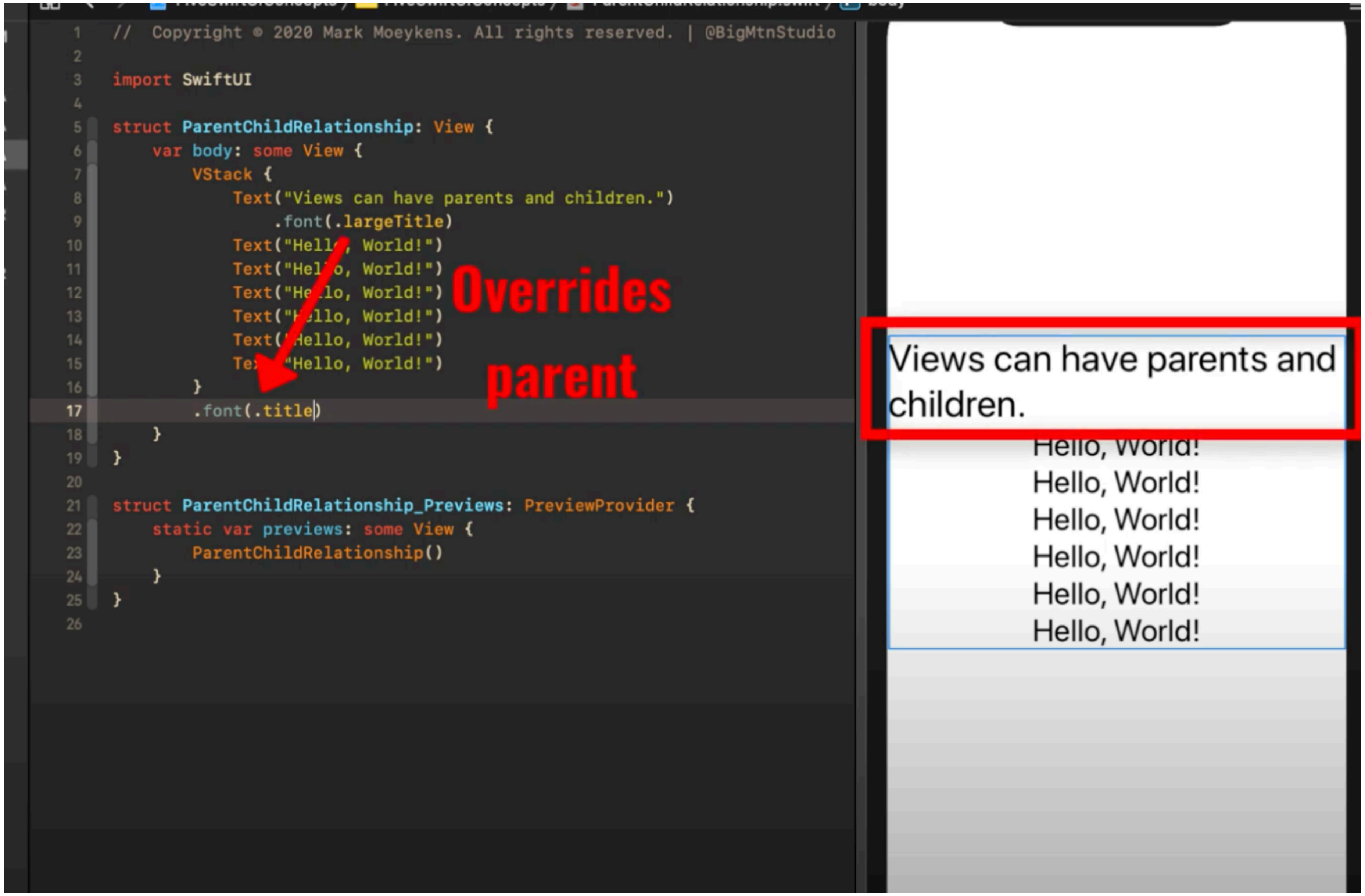
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!
Hello, World!



FiveSwiftUIConcepts / FiveSwiftUIConcepts / ParentChildRelationship.swift / body

```swift
//  Copyright © 2020 Mark Moeykens. All rights reserved. | @BigMtnStudio

import SwiftUI

struct ParentChildRelationship: View {
    var body: some View {
        VStack {
            Text("Views can have parents and children.")
                .font(.largeTitle)
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
        }
        .font(.title)
    }
}

struct ParentChildRelationship_Previews: PreviewProvider {
    static var previews: some View {
        ParentChildRelationship()
    }
}

```

```
Views can have parents and
children.
            Hello, World!
            Hello, World!
            Hello, World!
            Hello, World!
            Hello, World!
            Hello, World!
```



```swift
//  Copyright © 2020 Mark Moeykens. All rights reserved. | @BigMtnStudio

import SwiftUI

struct ParentChildRelationship: View {
    var body: some View {
        VStack {
            Text("Views can have parents and children.")
                .font(.largeTitle)
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
            Text("Hello, World!")
        }
        .font(.title)
    }
}

struct ParentChildRelationship_Previews: PreviewProvider {
    static var previews: some View {
        ParentChildRelationship()
    }
}
```

Overrides parent

Views can have parents and children.

Hello, World!

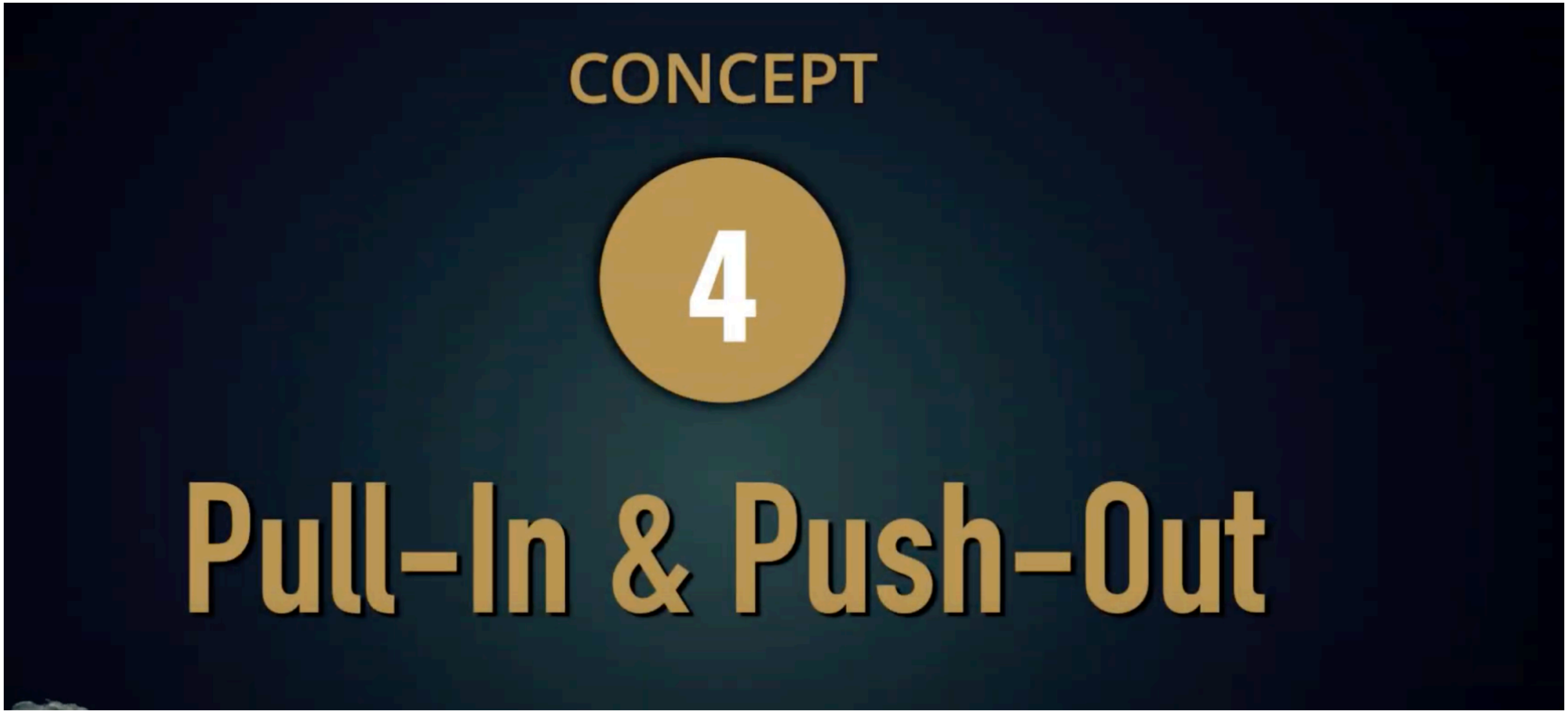
Hello, World!

Hello, World!

Hello, World!

Hello, World!

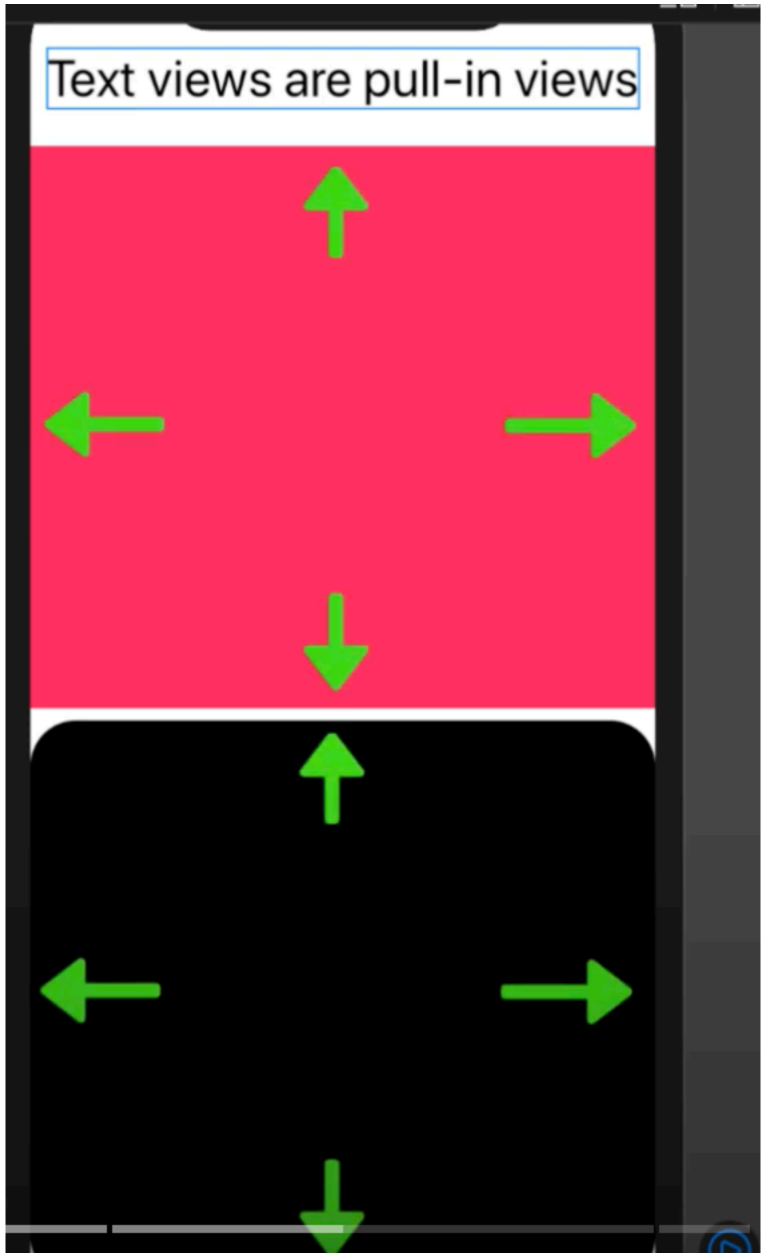
Hello, World!



CONCEPT

4

Pull-In & Push-Out

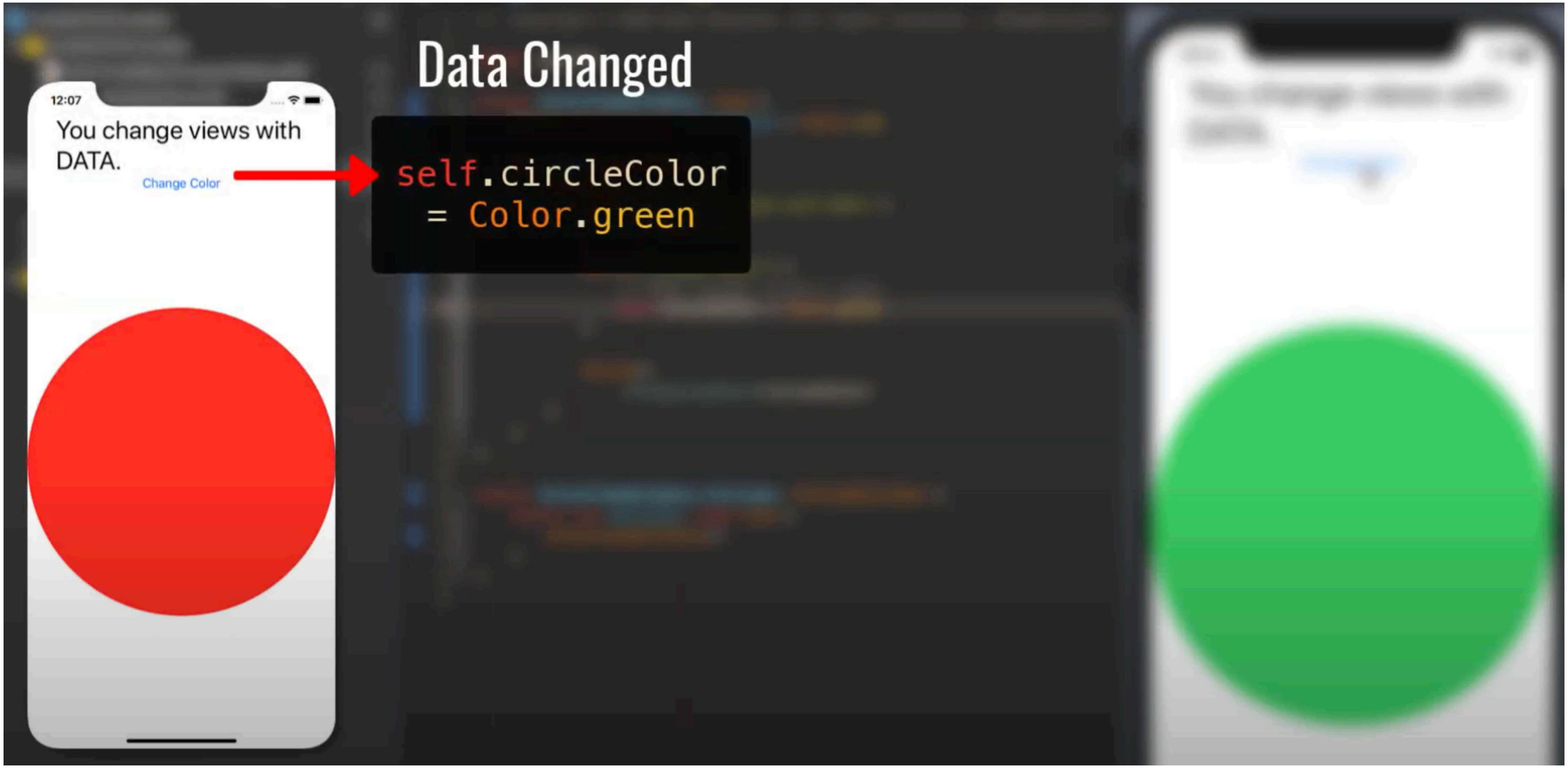


Text views are pull-in views

CONCEPT

5

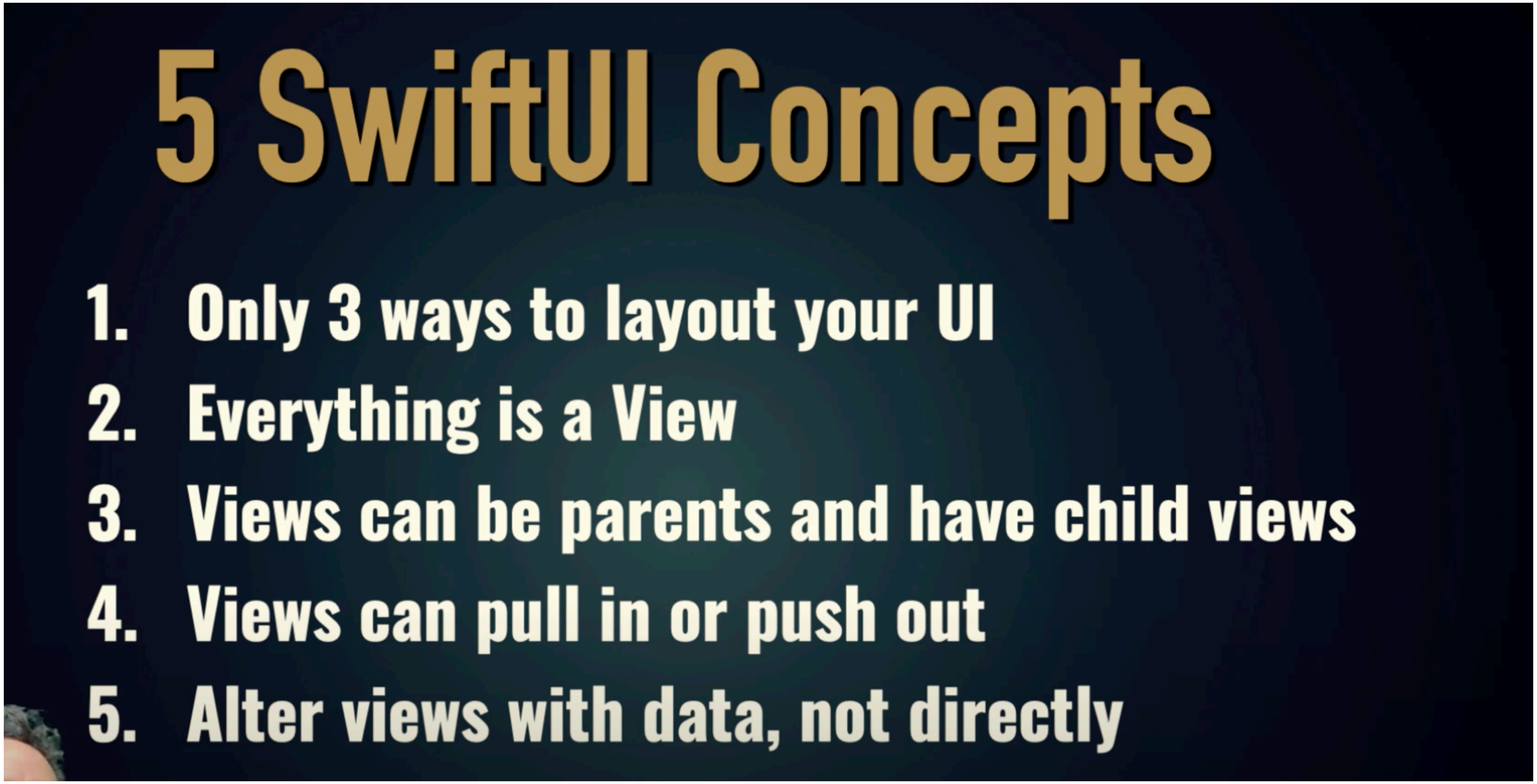
Change Views with DATA



Data Changed

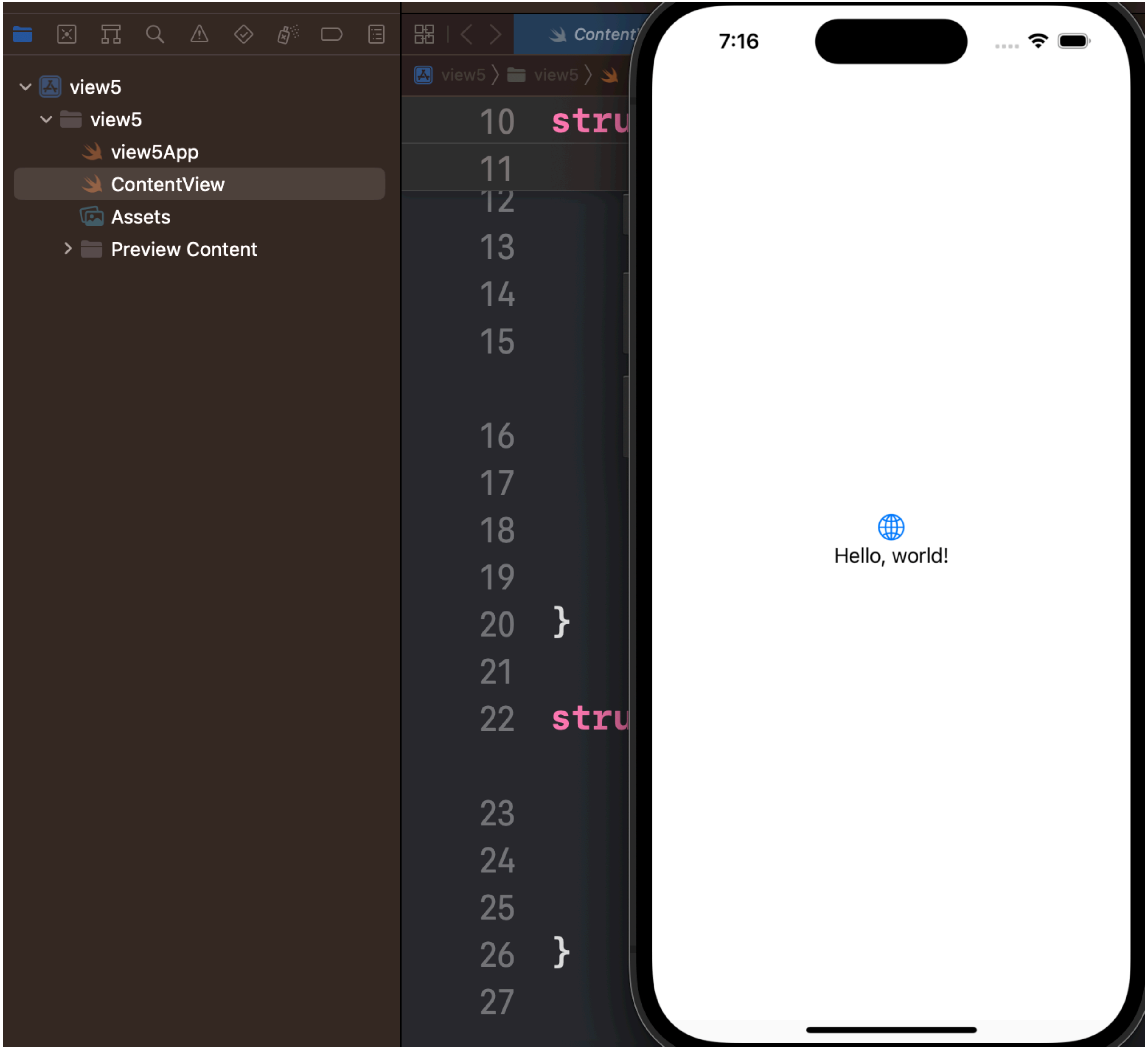
You change views with DATA.

```
self.circleColor
  = Color.green
```

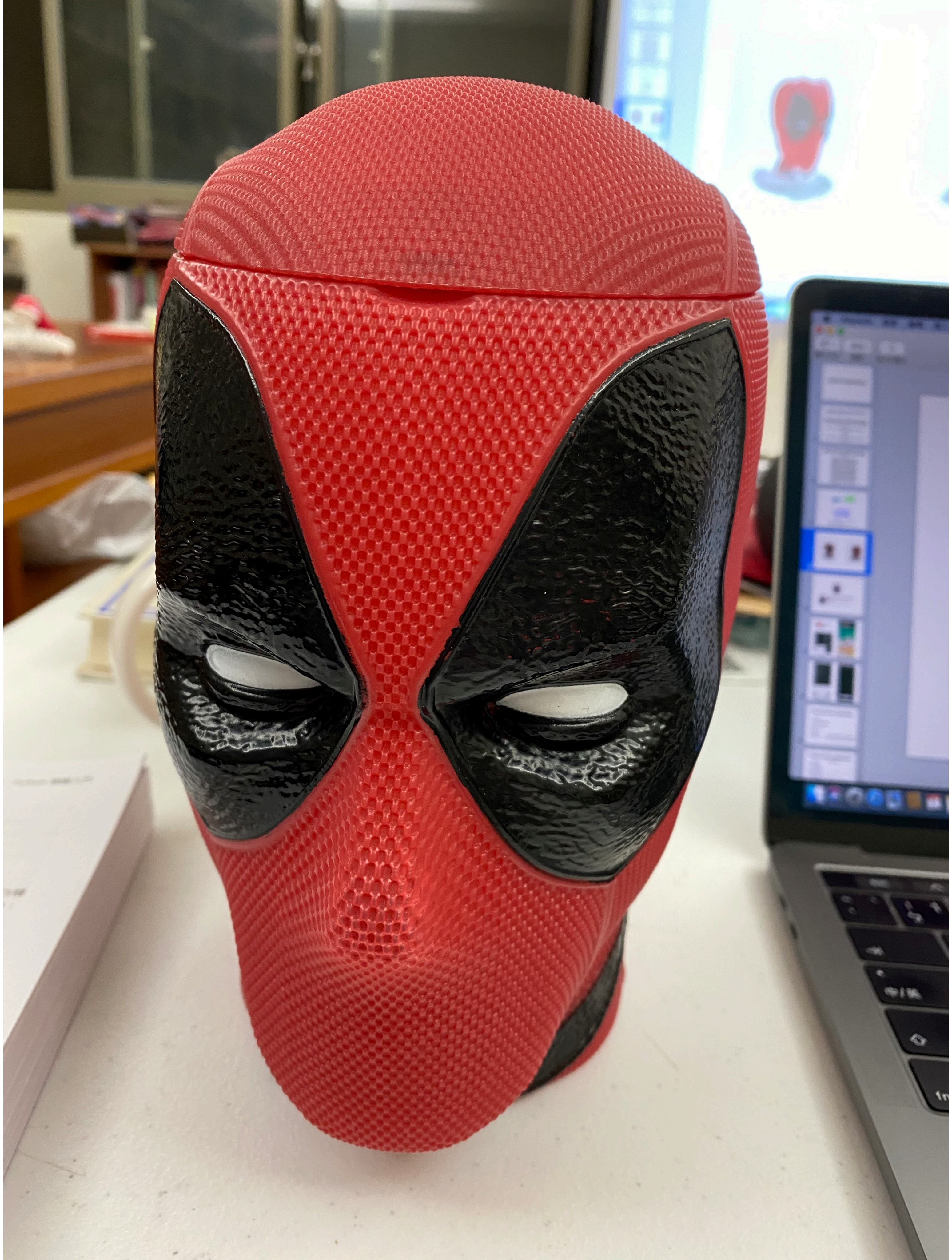


5 SwiftUI Concepts

1. Only 3 ways to layout your UI
2. Everything is a View
3. Views can be parents and have child views
4. Views can pull in or push out
5. Alter views with data, not directly


view5
view5
view5App
ContentView
Assets
Preview Content
Content
view5
view5
7:16
Hello, world!

```
stru

}

stru

}
```



教育人員將 Swift 納入課程

並非只有開發者體驗過 Swift 的潛力，世界各地的大學與學術機構都在 Mac 上使用 Swift 和 Xcode 進行教學，使學生可以運用最佳工具，擁有打造精彩 app 的能力。再者，透過 Apple 的「使用 Swift 開發」免額外付費課程，更能輕易從入門編碼轉換到使用 Swift 開發 app。

將 Swift 納入課程的大專校院

阿伯里斯特威斯大學 (Aberystwyth University)

曼哈頓區社區學院 (Borough of Manhattan Community College)

加州理工州立大學 (California Polytechnic State University)

中皮德蒙特社區學院 (Central Piedmont Community College)

山麓學院 (Foothill College)

福賽大學 (Full Sail University)

休士頓社區學院系統 (Houston Community College System)

Ingésup

勞森州立社區學院 (Lawson State Community College)

梅薩社區學院 (Mesa Community College)

西北堪薩斯技術學院 (Northwest Kansas Technical College)

普利茅斯大學 (Plymouth University)

皇家墨爾本理工大學 (RMIT University)

南方衛理公會大學 (Southern Methodist University)

史丹福大學 (Stanford University)

慕尼黑工業大學 (Technical University of Munich)

蒙特雷科技大學 (Tecnológico de Monterrey)

加州大學聖克魯斯分校 (University of California, Santa Cruz)

Swift

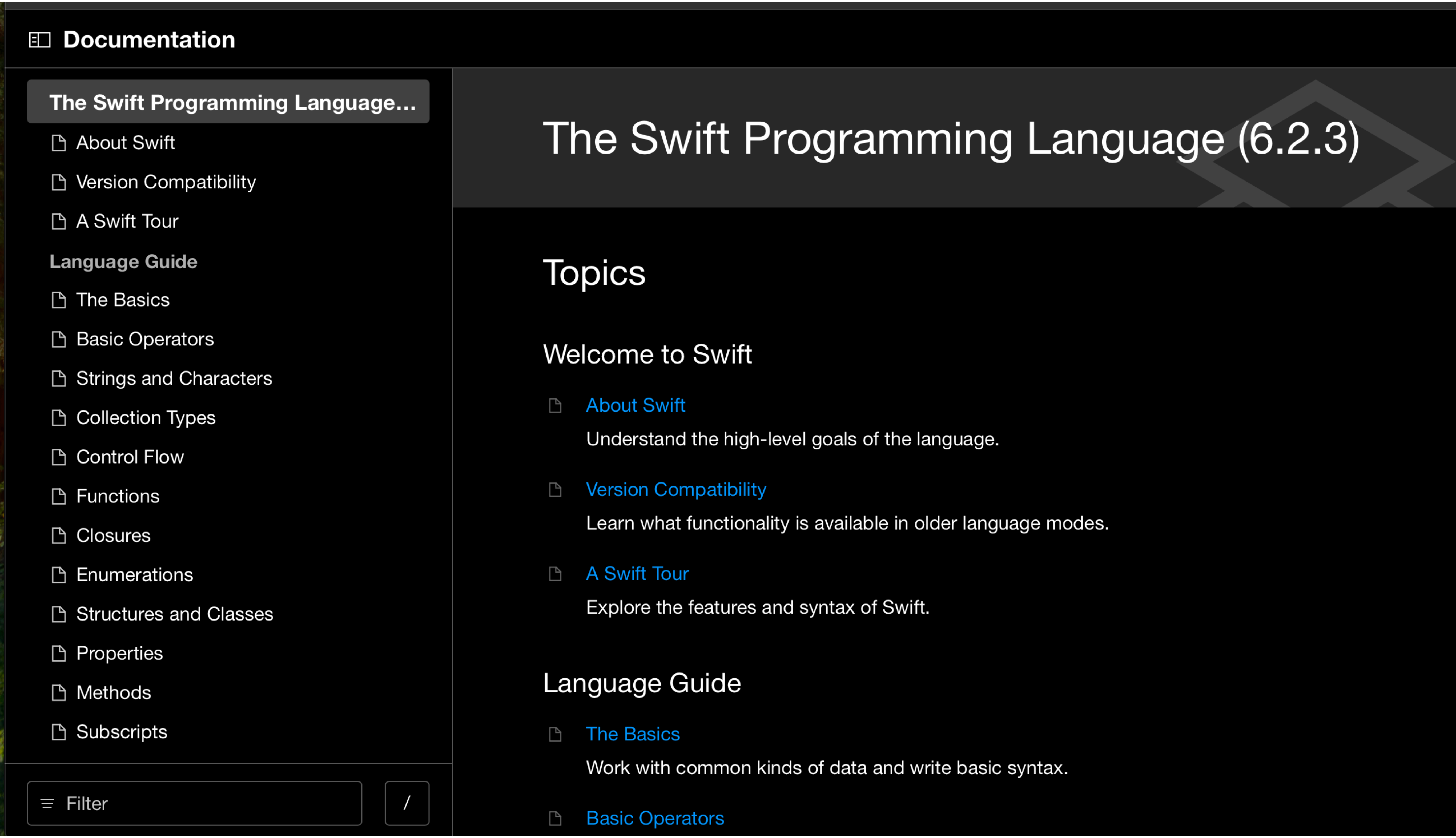
THE SWIFT PROGRAMMING LANGUAGE

SWIFT 5.4

The Swift Programming Language

Swift 5.3 Edition

2026.02.23



The Swift Programming Language (6.2.3)

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

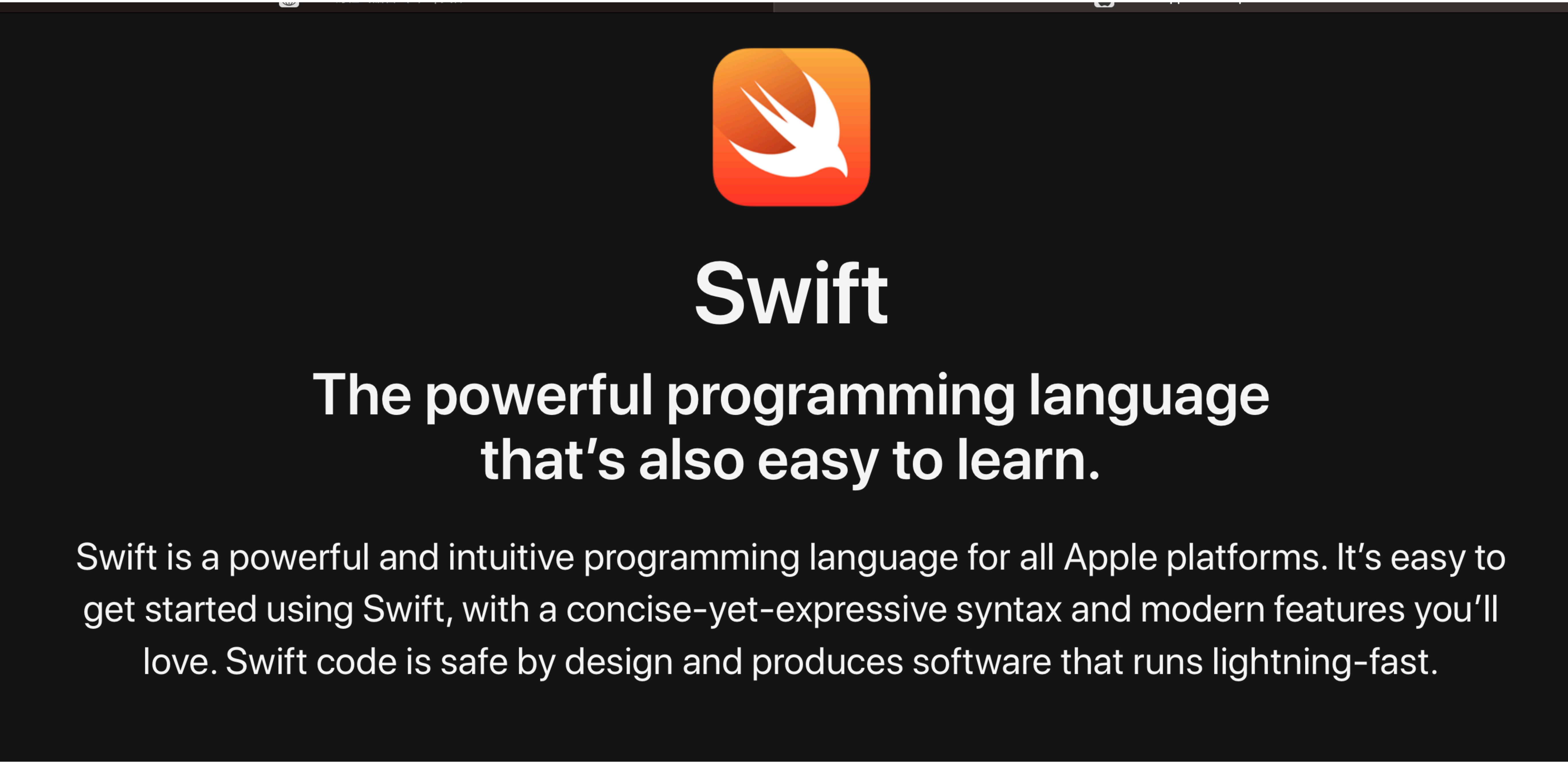
Explore the features and syntax of Swift.

Language Guide

The Basics

Work with common kinds of data and write basic syntax.

Basic Operators

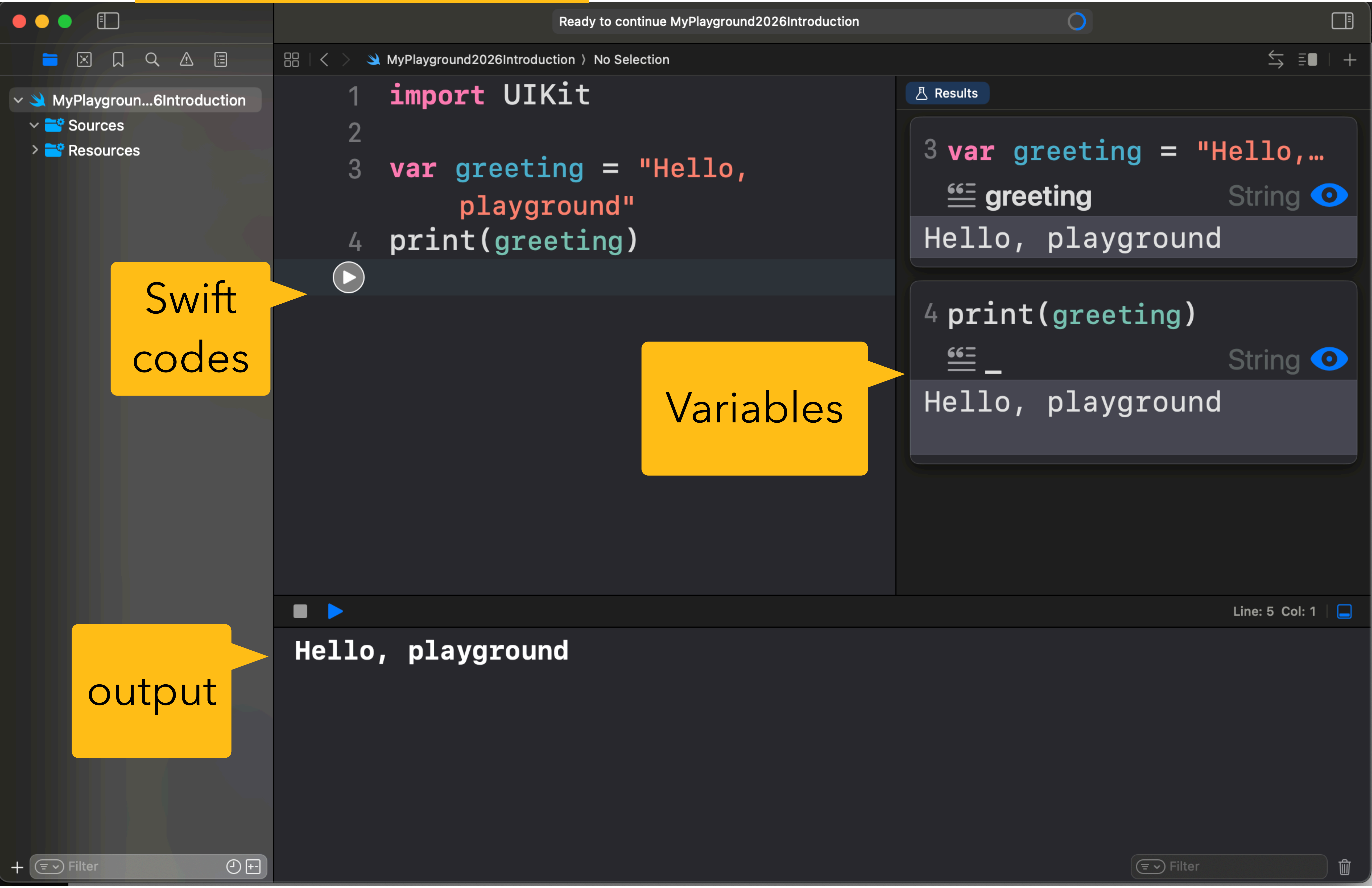


Swift

The powerful programming language that’s also easy to learn.

Swift is a powerful and intuitive programming language for all Apple platforms. It’s easy to get started using Swift, with a concise-yet-expressive syntax and modern features you’ll love. Swift code is safe by design and produces software that runs lightning-fast.

Create a playground



Ready to continue MyPlayground2026Introduction
MyPlayground2026Introduction 〉 No Selection
MyPlaygroun…6Introduction
Sources
Resources
Results
Line: 5 Col: 1
Filter
Filter

```
import UIKit

var greeting = "Hello,
    playground"
print(greeting)
```

```
3 var greeting = "Hello,…
greeting    String
Hello, playground

4 print(greeting)
_    String
Hello, playground
```

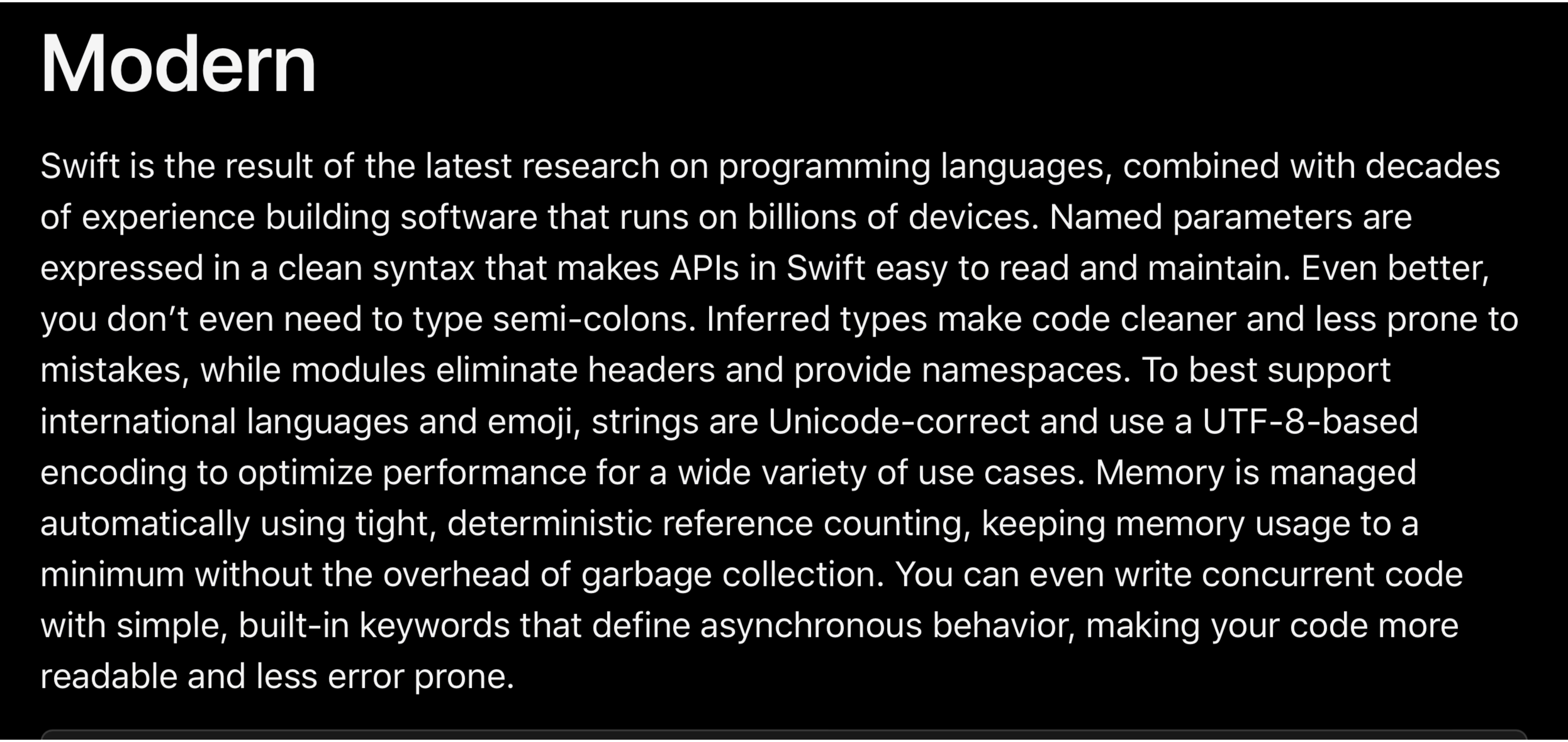
```
Hello, playground
```

Swift codes

Variables

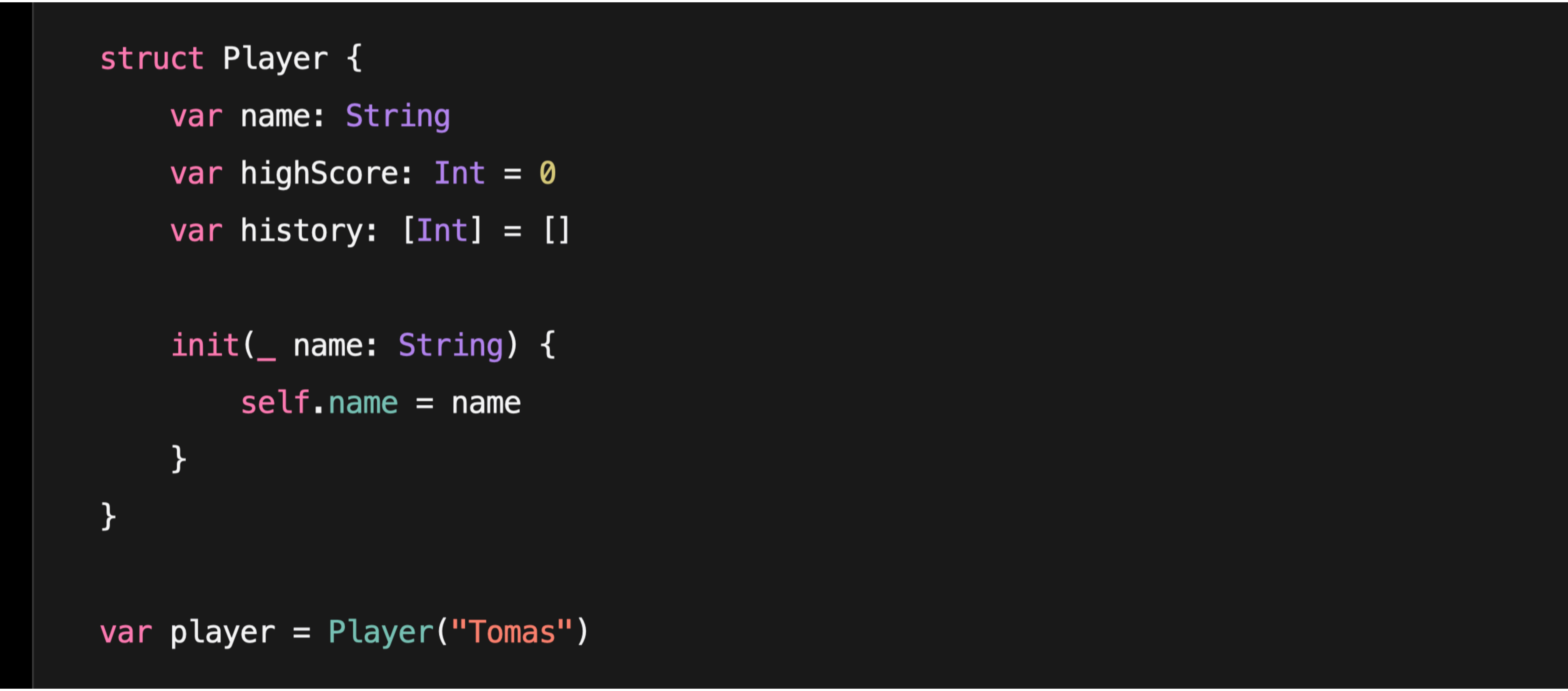
output

Modern



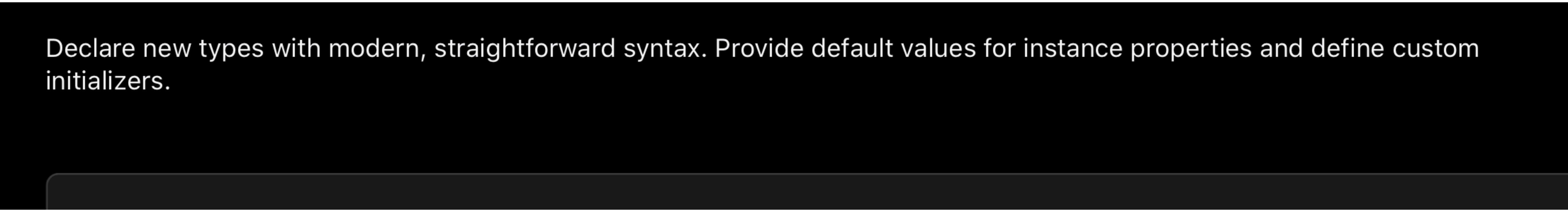
Modern

Swift is the result of the latest research on programming languages, combined with decades of experience building software that runs on billions of devices. Named parameters are expressed in a clean syntax that makes APIs in Swift easy to read and maintain. Even better, you don’t even need to type semi-colons. Inferred types make code cleaner and less prone to mistakes, while modules eliminate headers and provide namespaces. To best support international languages and emoji, strings are Unicode-correct and use a UTF-8-based encoding to optimize performance for a wide variety of use cases. Memory is managed automatically using tight, deterministic reference counting, keeping memory usage to a minimum without the overhead of garbage collection. You can even write concurrent code with simple, built-in keywords that define asynchronous behavior, making your code more readable and less error prone.

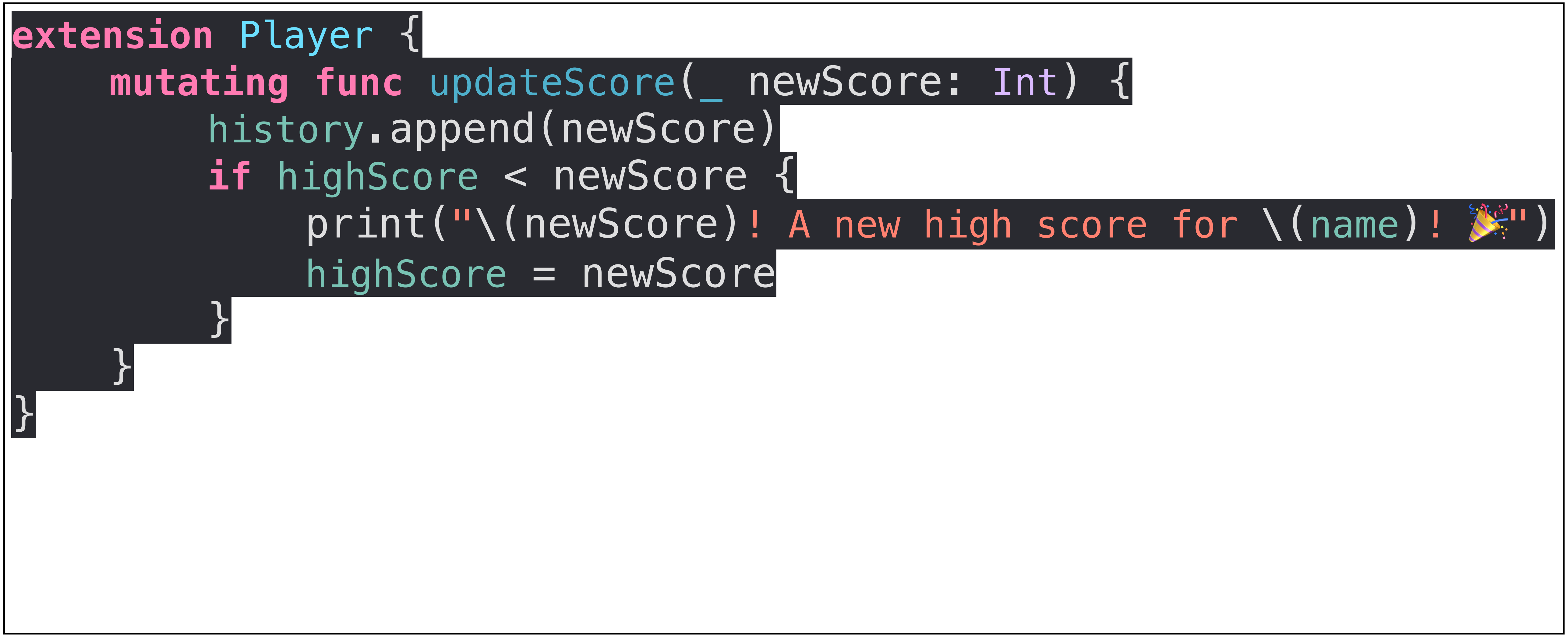


```swift
struct Player {
    var name: String
    var highScore: Int = 0
    var history: [Int] = []

    init(_ name: String) {
        self.name = name
    }
}

var player = Player("Tomas")
```

Declare new types with modern, straightforward syntax. Provide default values for instance properties and define custom initializers.

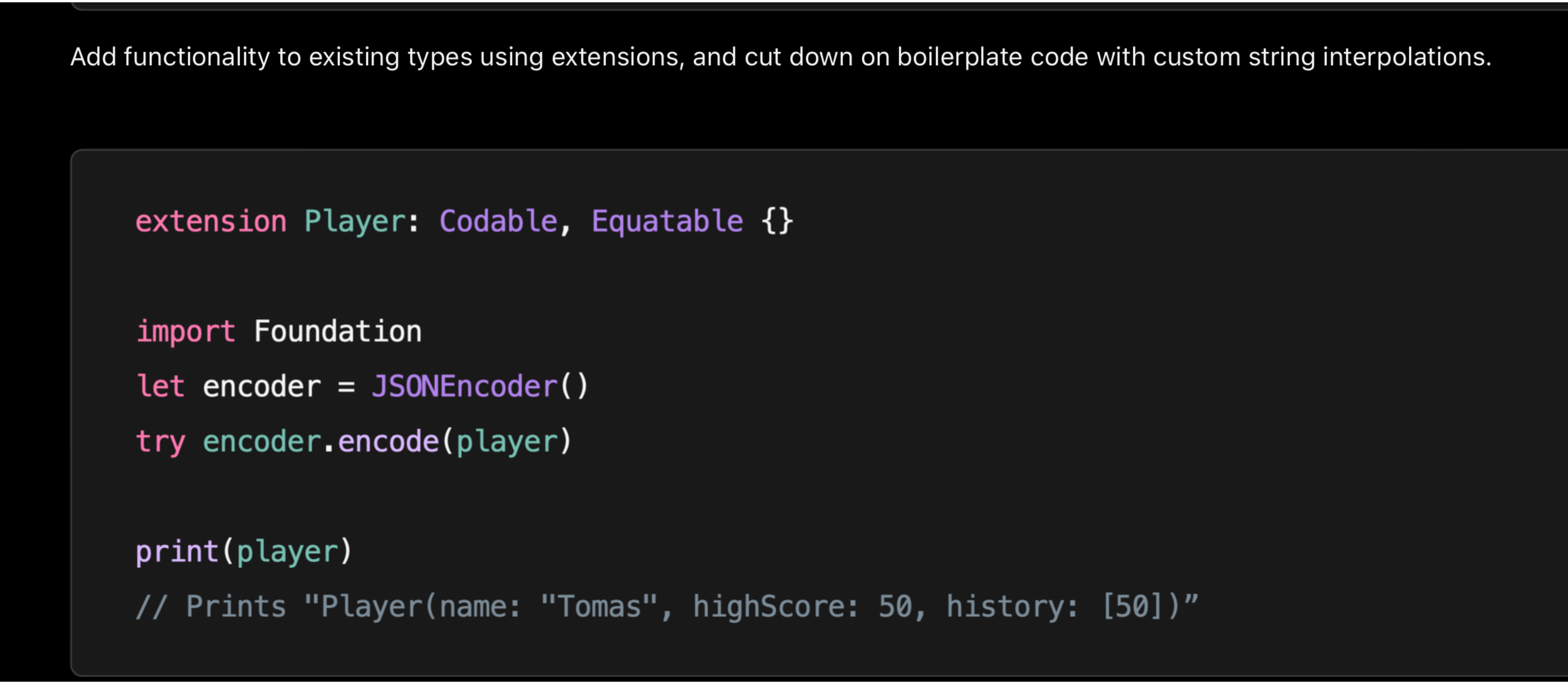


Declare new types with modern, straightforward syntax. Provide default values for instance properties and define custom initializers.



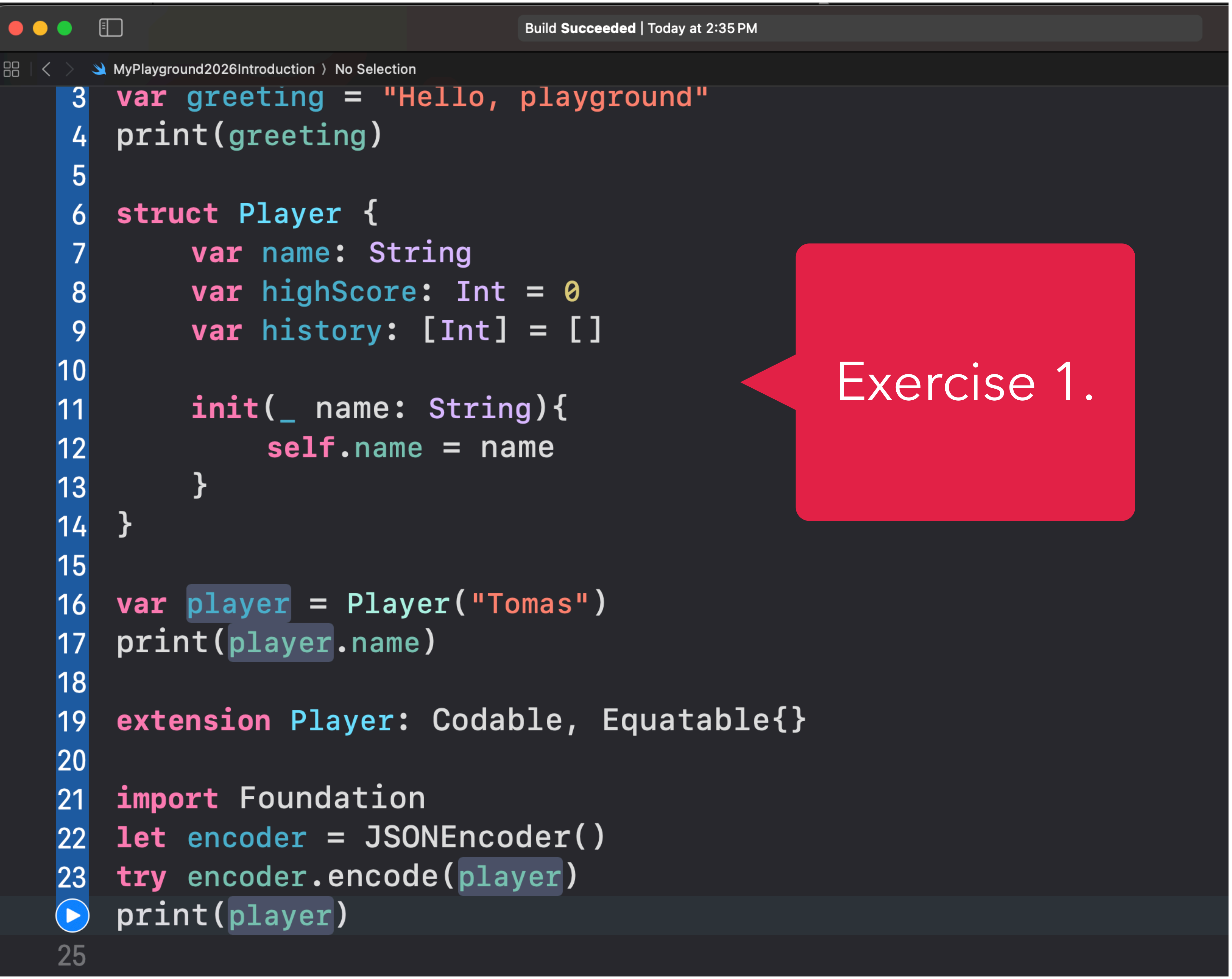
```swift
extension Player {
    mutating func updateScore(_ newScore: Int) {
        history.append(newScore)
        if highScore < newScore {
            print("\(newScore)! A new high score for \(name)! 🎉")
            highScore = newScore
        }
    }
}
```

Quickly extend your custom types to take advantage of powerful language features, such as automatic JSON encoding and decoding.



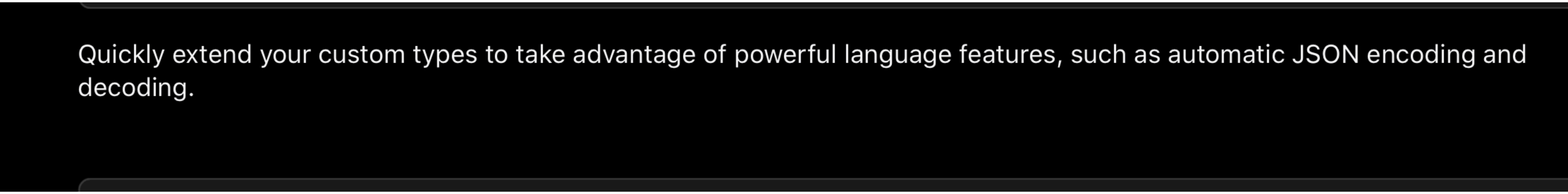
Add functionality to existing types using extensions, and cut down on boilerplate code with custom string interpolations.

```swift
extension Player: Codable, Equatable {}

import Foundation
let encoder = JSONEncoder()
try encoder.encode(player)

print(player)
// Prints "Player(name: "Tomas", highScore: 50, history: [50])”
```

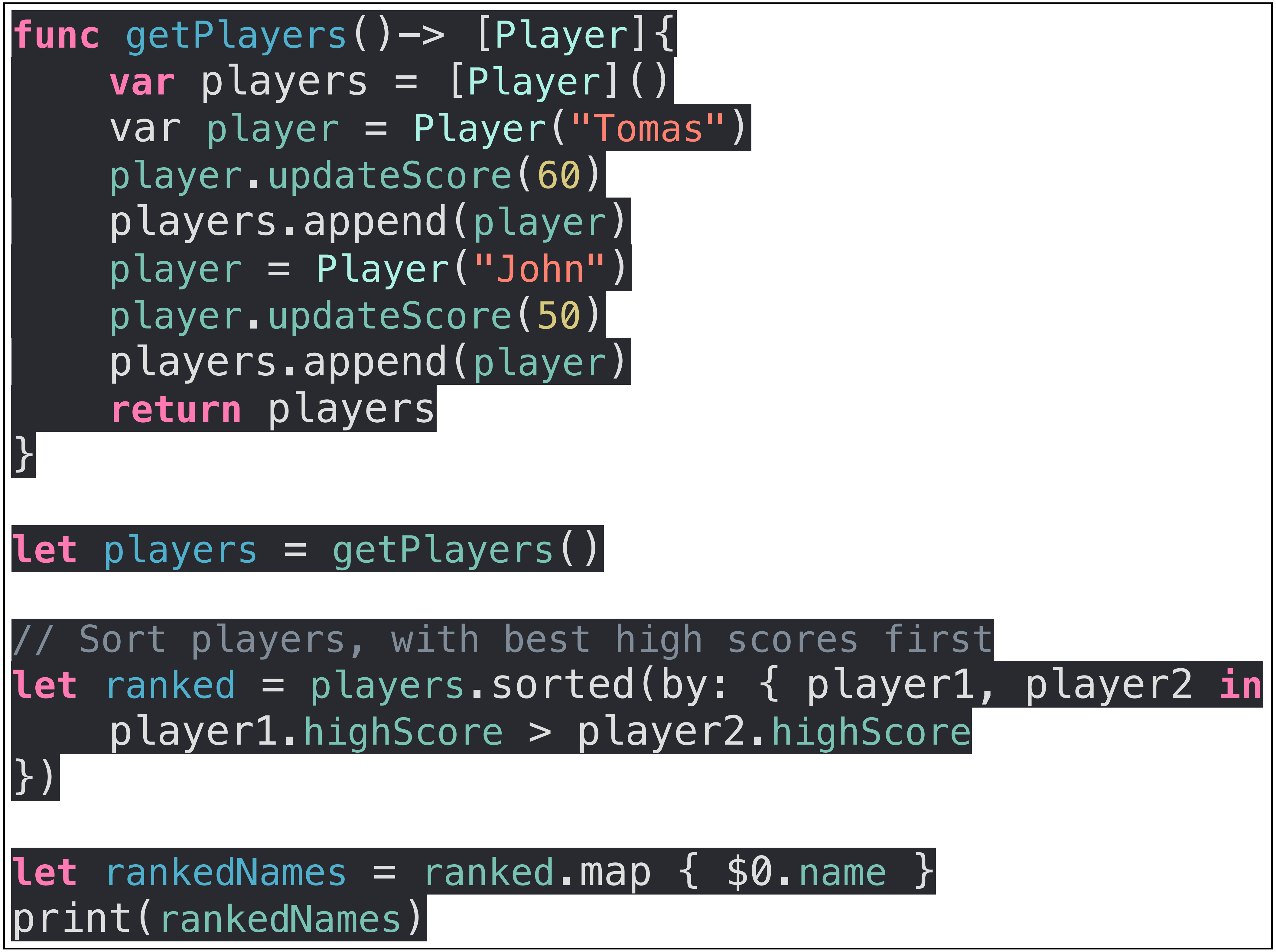

Build Succeeded | Today at 2:35 PM
MyPlayground2026Introduction 〉 No Selection

```swift
var greeting = "Hello, playground"
print(greeting)

struct Player {
    var name: String
    var highScore: Int = 0
    var history: [Int] = []

    init(_ name: String){
        self.name = name
    }
}

var player = Player("Tomas")
print(player.name)

extension Player: Codable, Equatable{}

import Foundation
let encoder = JSONEncoder()
try encoder.encode(player)
print(player)
```

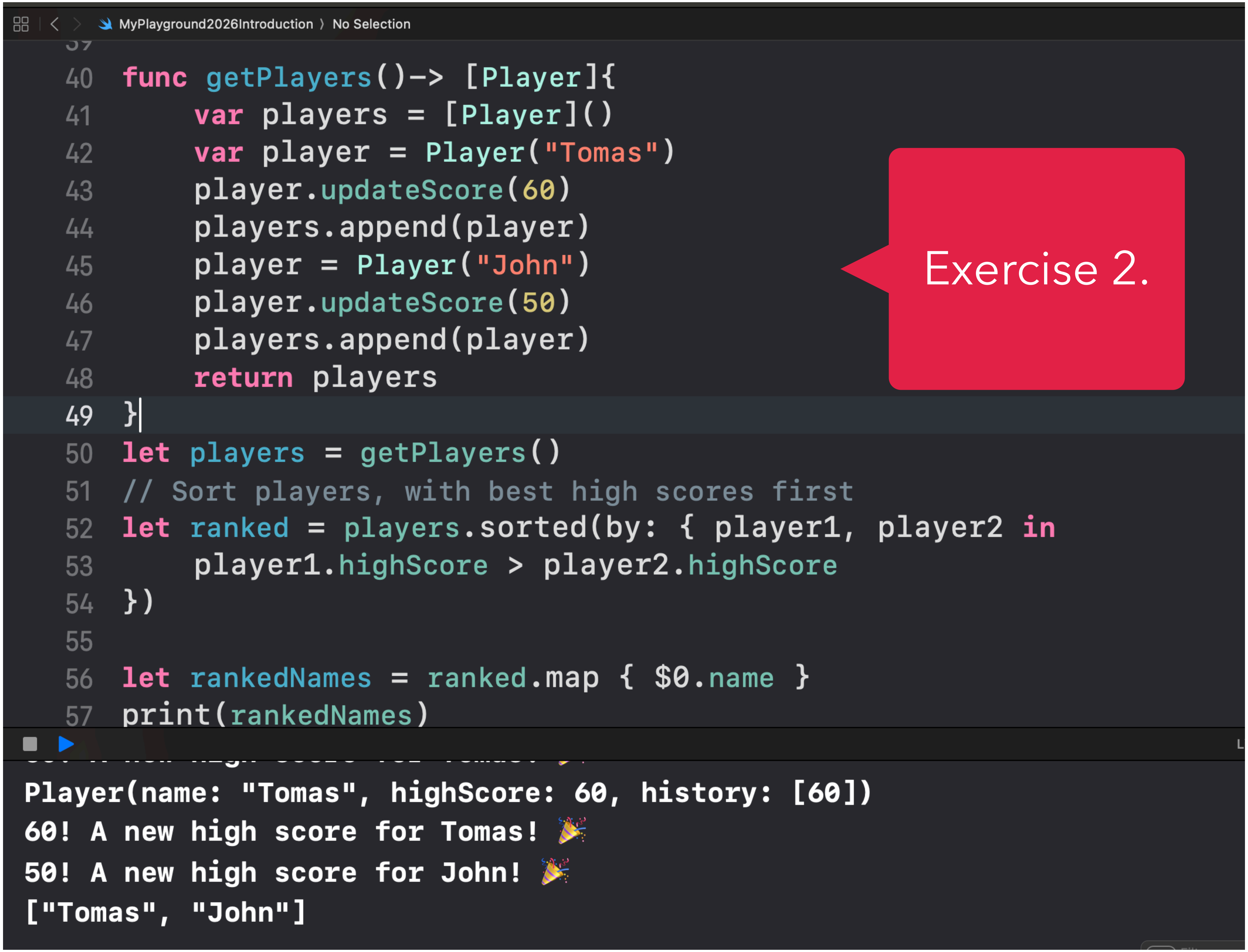
Exercise 1.



Quickly extend your custom types to take advantage of powerful language features, such as automatic JSON encoding and decoding.



```swift
func getPlayers()-> [Player]{
    var players = [Player]()
    var player = Player("Tomas")
    player.updateScore(60)
    players.append(player)
    player = Player("John")
    player.updateScore(50)
    players.append(player)
    return players
}

let players = getPlayers()

// Sort players, with best high scores first
let ranked = players.sorted(by: { player1, player2 in
    player1.highScore > player2.highScore
})

let rankedNames = ranked.map { $0.name }
print(rankedNames)
```



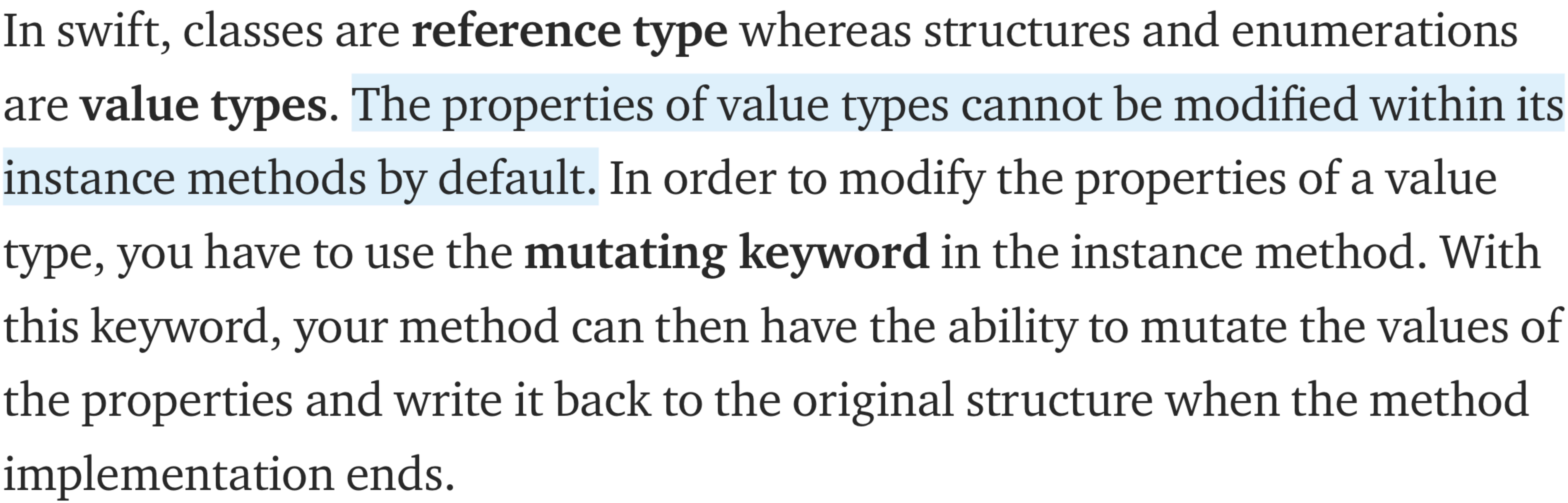
```swift
func getPlayers()-> [Player]{
    var players = [Player]()
    var player = Player("Tomas")
    player.updateScore(60)
    players.append(player)
    player = Player("John")
    player.updateScore(50)
    players.append(player)
    return players
}
let players = getPlayers()
// Sort players, with best high scores first
let ranked = players.sorted(by: { player1, player2 in
    player1.highScore > player2.highScore
})

let rankedNames = ranked.map { $0.name }
print(rankedNames)
```

```
Player(name: "Tomas", highScore: 60, history: [60])
60! A new high score for Tomas! 🎉
50! A new high score for John! 🎉
["Tomas", "John"]
```

Exercise 2.

Mutating

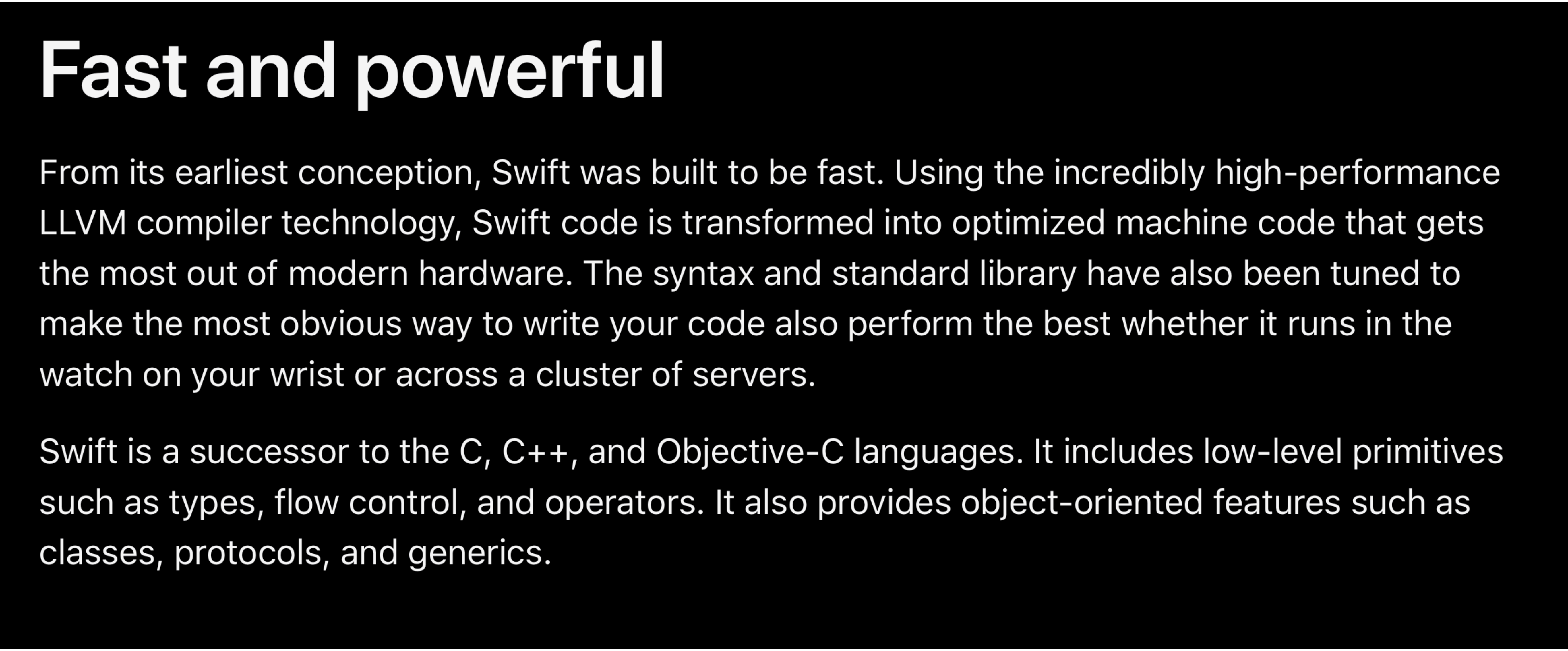
https://medium.com/the-andela-way/swift-understanding-mutating-functions-in-two-minutes-d9e363904e3a



In swift, classes are **reference type** whereas structures and enumerations are **value types**. The properties of value types cannot be modified within its instance methods by default. In order to modify the properties of a value type, you have to use the **mutating keyword** in the instance method. With this keyword, your method can then have the ability to mutate the values of the properties and write it back to the original structure when the method implementation ends.

```
struct Stack {
    public private(set) var items = [Int]() // Empty items array

    mutating func push(_ item: Int) {
        items.append(item)
    }

    mutating func pop() -> Int? {
        if !items.isEmpty {
           return items.removeLast()
        }
        return nil
    }
}


var stack = Stack()
stack.push(4)
stack.push(78)
stack.items // [4, 78]
stack.pop()
stack.items // [4]
```

Exercise 3.



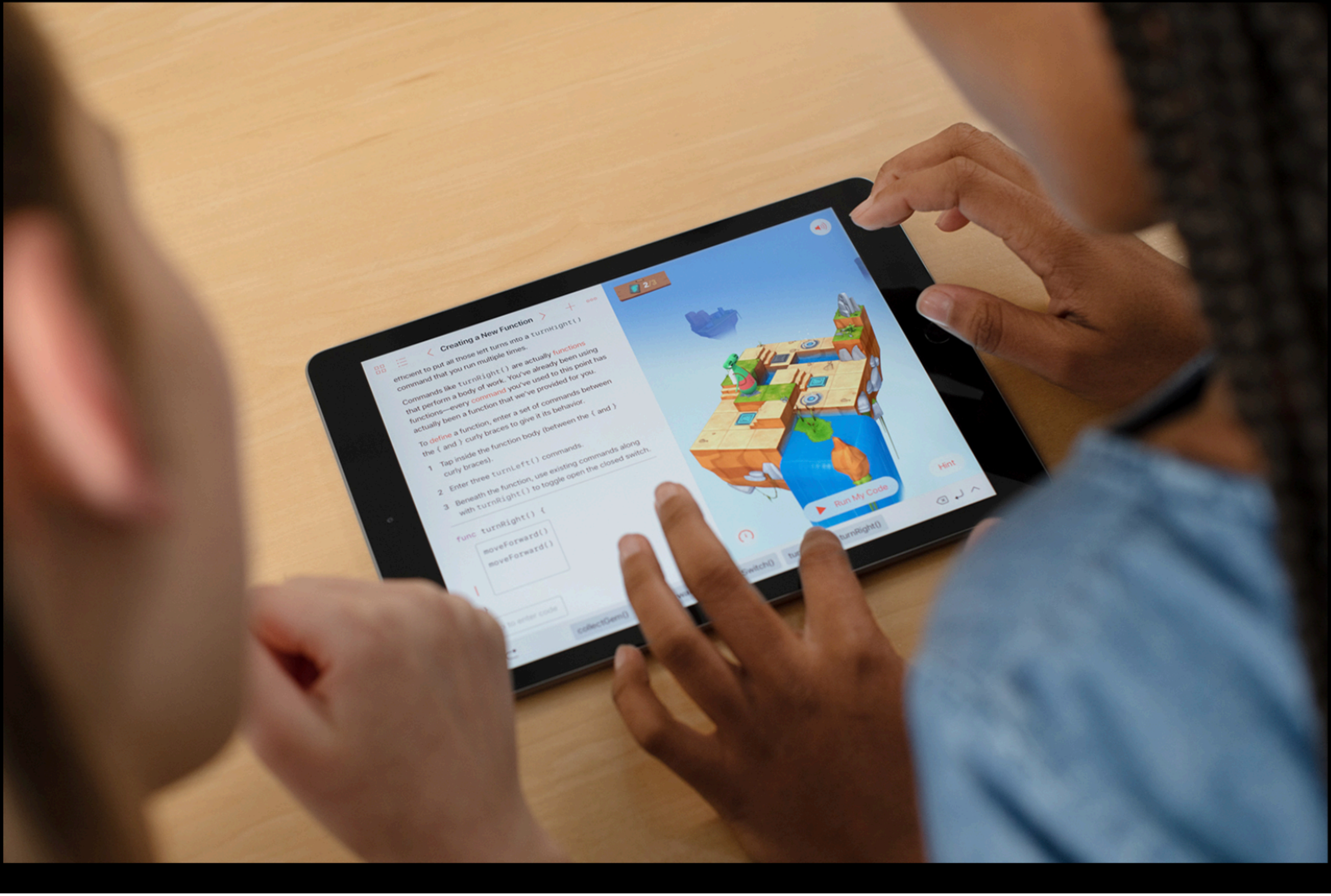
Fast and powerful

From its earliest conception, Swift was built to be fast. Using the incredibly high-performance LLVM compiler technology, Swift code is transformed into optimized machine code that gets the most out of modern hardware. The syntax and standard library have also been tuned to make the most obvious way to write your code also perform the best whether it runs in the watch on your wrist or across a cluster of servers.

Swift is a successor to the C, C++, and Objective-C languages. It includes low-level primitives such as types, flow control, and operators. It also provides object-oriented features such as classes, protocols, and generics.

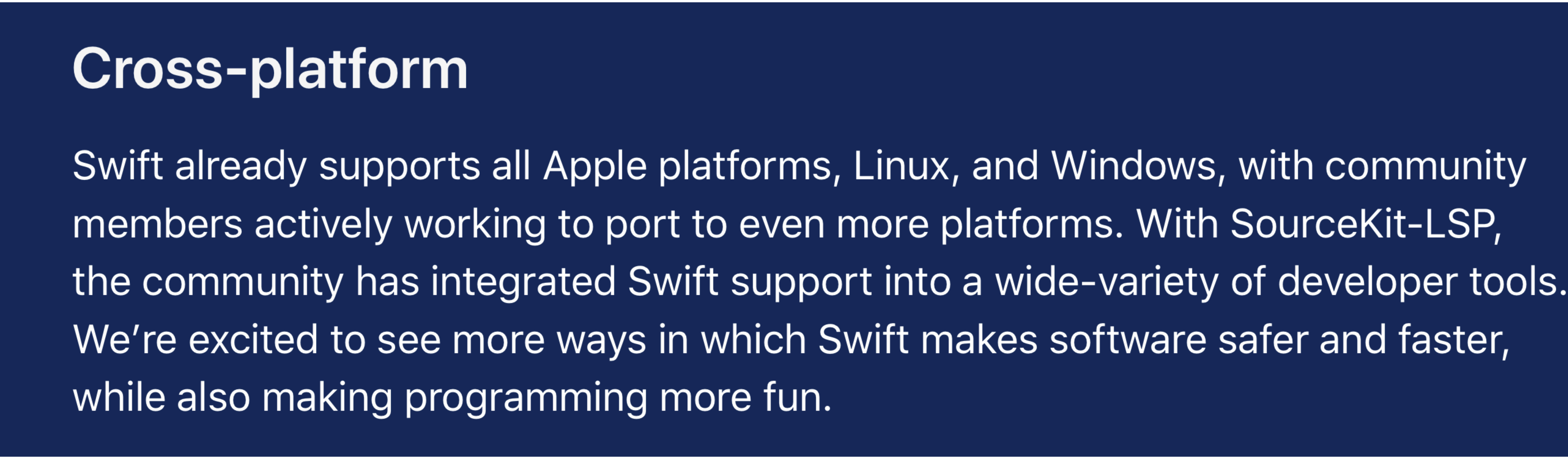
Great first language

Swift can open doors to the world of coding. In fact, it was designed to be anyone's first programming language, whether you're still in school or exploring new career paths. For educators, Apple created free curriculum to teach Swift both in and out of the classroom. First-time coders can download Swift Playground — an app for iPad and Mac that makes getting started with Swift code interactive and fun.



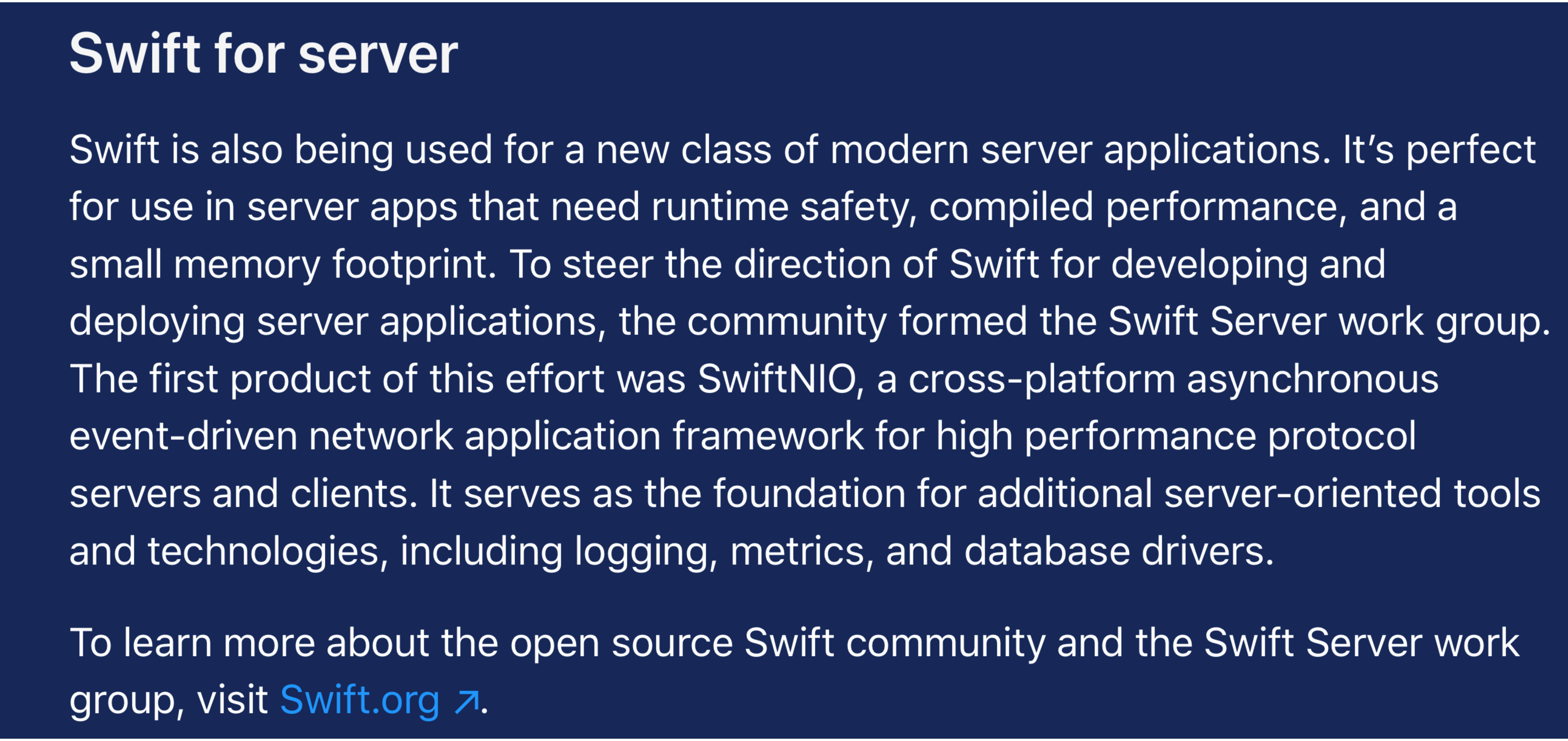
Open source

Swift is developed in the open at Swift.org, with source code, a bug tracker, forums, and regular development builds available for everyone. This broad community of developers, both inside Apple as well as hundreds of outside contributors, work together to make Swift even more amazing. There is an even broader range of blogs, podcasts, conferences, and meetups where developers in the community share their experiences of how best to use Swift.



Cross-platform

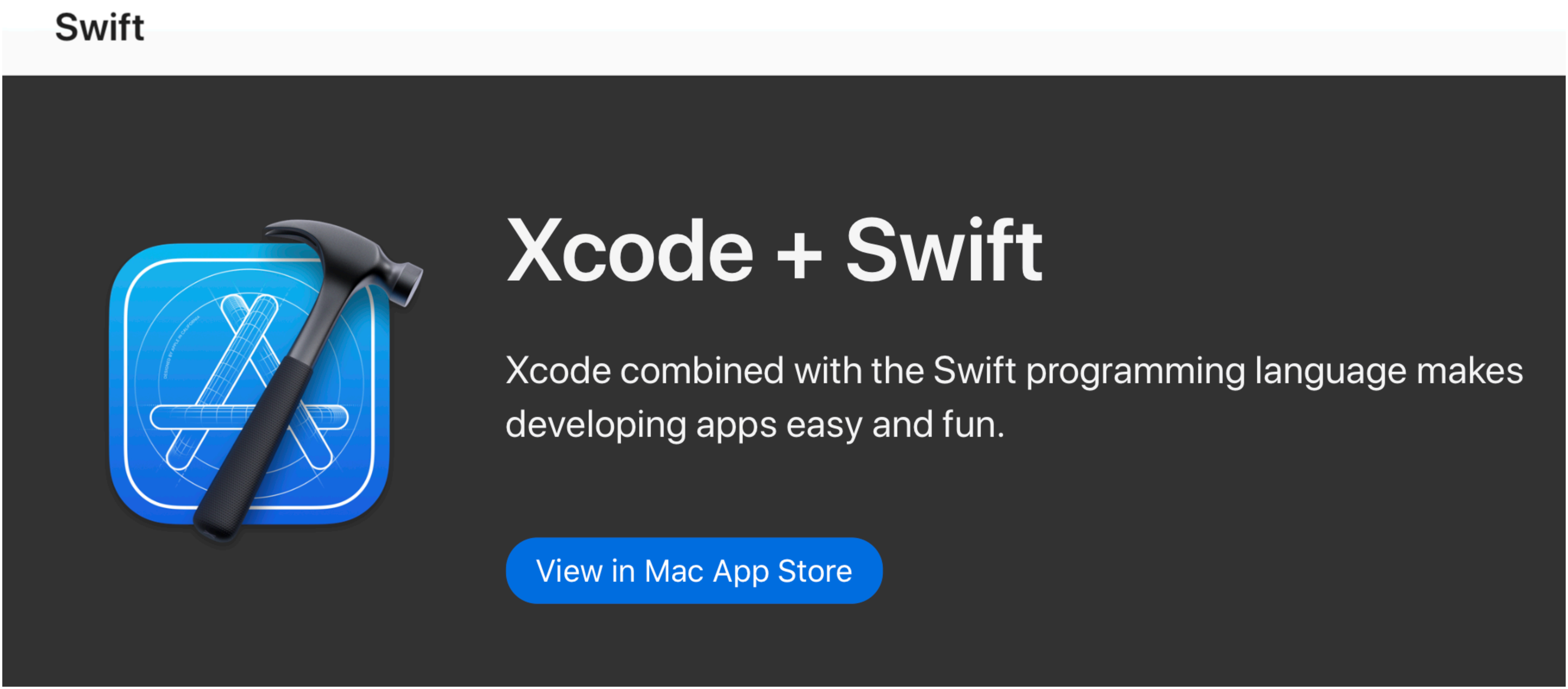
Swift already supports all Apple platforms, Linux, and Windows, with community members actively working to port to even more platforms. With SourceKit-LSP, the community has integrated Swift support into a wide-variety of developer tools. We’re excited to see more ways in which Swift makes software safer and faster, while also making programming more fun.



Swift for server

Swift is also being used for a new class of modern server applications. It’s perfect for use in server apps that need runtime safety, compiled performance, and a small memory footprint. To steer the direction of Swift for developing and deploying server applications, the community formed the Swift Server work group. The first product of this effort was SwiftNIO, a cross-platform asynchronous event-driven network application framework for high performance protocol servers and clients. It serves as the foundation for additional server-oriented tools and technologies, including logging, metrics, and database drivers.

To learn more about the open source Swift community and the Swift Server work group, visit Swift.org ↗.



Swift

Xcode + Swift

Xcode combined with the Swift programming language makes developing apps easy and fun.

Porgramming Language I

- 1951 - Regional Assembly Language
- 1952 - Autocode
- 1954 - FORTRAN **
- 1954 - IPL (LISP的先驅)
- 1955 - FLOW-MATIC (COBOL的先驅)
- 1957 - COMTRAN (COBOL的先驅)
- 1958 - LISP **
- 1958 - ALGOL 58
- 1959 - FACT (COBOL的先驅)
- 1959 - COBOL **
- 1962 - APL
- 1962 - Simula
- 1962 - SNOBOL
- 1963 - CPL (C的先驅)
- 1964 - BASIC
- 1964 - PL/I
- 1967 - BCPL (C的先驅)
-

**有三個現代程式語言於1950年代被設計出來

這三者所衍生的語言直到今日仍舊廣泛地被採用

Porgramming Language II

確立了基礎範式

- 1968 - Logo
- 1970 - Pascal
- 1970 - Forth
- 1972 - C語言
- 1972 - Smalltalk
- 1972 - Prolog
- 1973 - ML
- 1975 - Scheme
- 1978 - SQL (起先只是一種查詢語言，擴充之後也具備了程式結構)
-

Porgramming Language III

1980年代：增強、模組、效能

- 1980 - Ada
- 1983 - C++ (就像有類別的C)
- 1984 - Common Lisp
- 1985 - Eiffel
- 1986 - Erlang
- 1987 - Perl
- 1988 - Tcl
- 1989 - FL (Backus)
-

C++合併了物件導向以及系統程式設計

Programming Languages for Internet

提升程式設計師的生產力

- 1990 - Haskell
- 1991 - Python
- 1991 - Visual Basic
- 1993 - Ruby
- 1993 - Lua
- 1994 - CLOS (part of ANSI Common Lisp)
- 1995 - Java
- 1995 - Delphi (Object Pascal)
- 1995 - JavaScript
- 1995 - PHP
- 1997 - REBOL
- 1999 - D
-

現今的趨勢

- 元件導向(component-oriented)軟體開發
- 更重視分散式及移動式的應用

- 2001 - C#
- 2001 - Visual Basic .NET
- 2002 - F#
- 2003 - Scala
- 2003 - Factor
- 2006 - Windows PowerShell
- 2007 - Clojure
- 2009 - Go
- 2014 - Swift (程式語言)
-

Mathematics, Statistics and AI computing

- Mathematical and Statistical Softwares
    - C, C++
    - R, SAS and MATLAB
    - Python
    - Swift **
- Mathematical AI and Applications
    - Parallel and distributed computing
    - Medical images
    - Apple App store
    - Integration of databases, mathematical models, neural networks to Apps on iMac and iphones **

Swift is a fantastic way to write software, whether it's for phones, desktops, servers, or anything else that runs code. It's a safe, fast, and interactive programming language that combines the best in modern language thinking with wisdom from the wider Apple engineering culture and the diverse contributions from its open-source community.

The compiler is optimized for performance and the language is optimized for development, without compromising on either.

Swift is friendly to new programmers. It’s an industrial-quality programming language that’s as expressive and enjoyable as a scripting language.

Writing Swift code in a playground lets you experiment with code and see the results immediately, without the overhead of building and running an app.

Swift defines away large classes of common programming errors by adopting modern programming patterns:

- Variables are always initialized before use.
- Array indices are checked for out-of-bounds errors.
- Integers are checked for overflow.

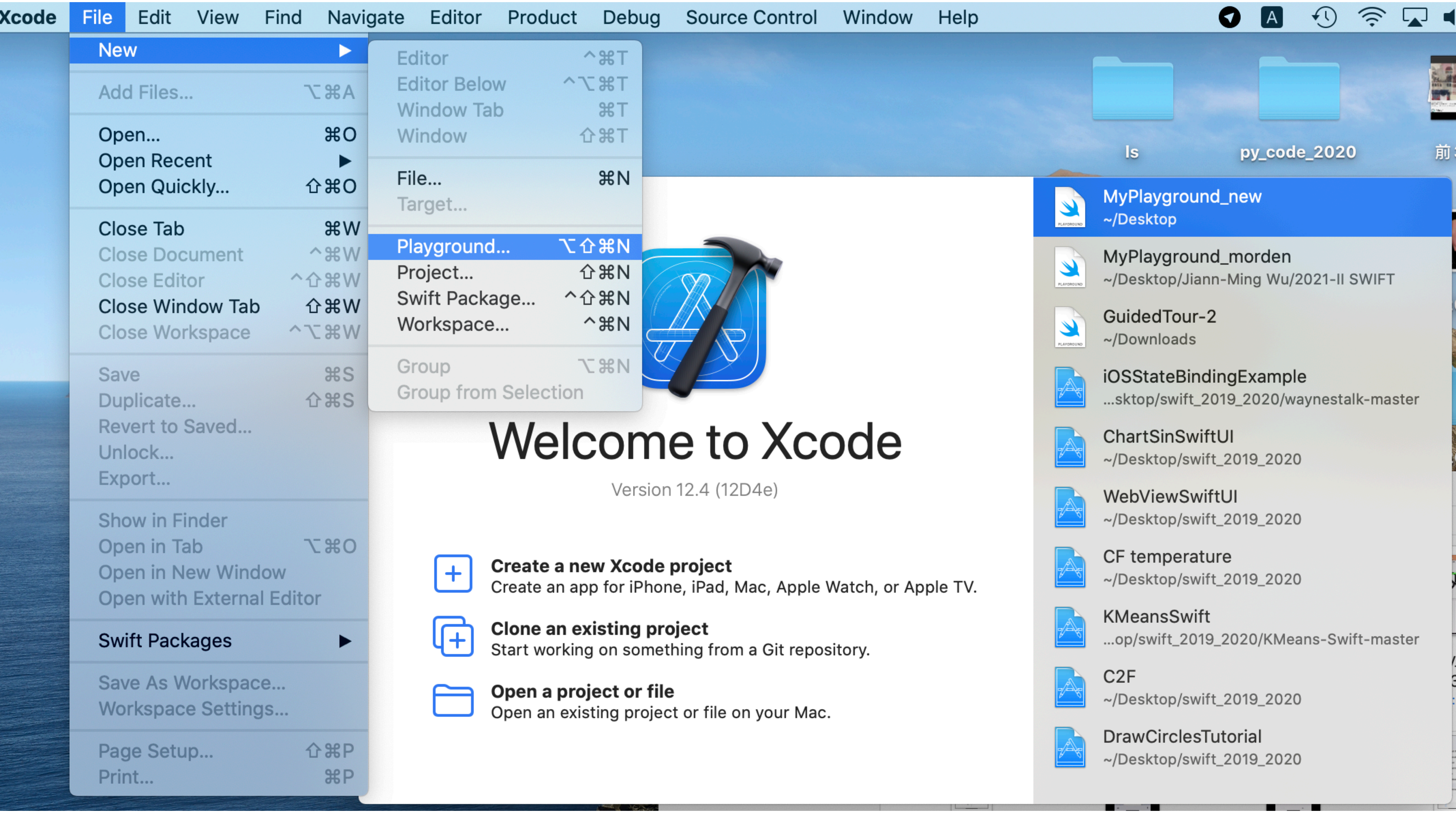
- Optionals ensure that `nil` values are handled explicitly.
- Memory is managed automatically.
- Error handling allows controlled recovery from unexpected failures.

Swift combines powerful type inference and pattern matching with a modern, lightweight syntax, allowing complex ideas to be expressed in a clear and concise manner. As a result, code is not just easier to write, but easier to read and maintain as well.

Swift code is compiled and optimized to get the most out of modern hardware. The syntax and standard library have been designed based on the guiding principle that the obvious way to write your code should also perform the best.

Its combination of safety and speed make Swift an excellent choice for everything from “Hello, world!” to an entire operating system.

Swift has been years in the making, and it continues to evolve with new features and capabilities. Our goals for Swift are ambitious. We can’t wait to see what you create with it.



Welcome to Xcode

Version 12.4 (12D4e)

Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.

Clone an existing project
Start working on something from a Git repository.

Open a project or file
Open an existing project or file on your Mac.

MyPlayground_new
~/Desktop

MyPlayground_morden
~/Desktop/Jiann-Ming Wu/2021-II SWIFT

GuidedTour-2
~/Downloads

iOSStateBindingExample
...sktop/swift_2019_2020/waynestalk-master

ChartSinSwiftUI
~/Desktop/swift_2019_2020

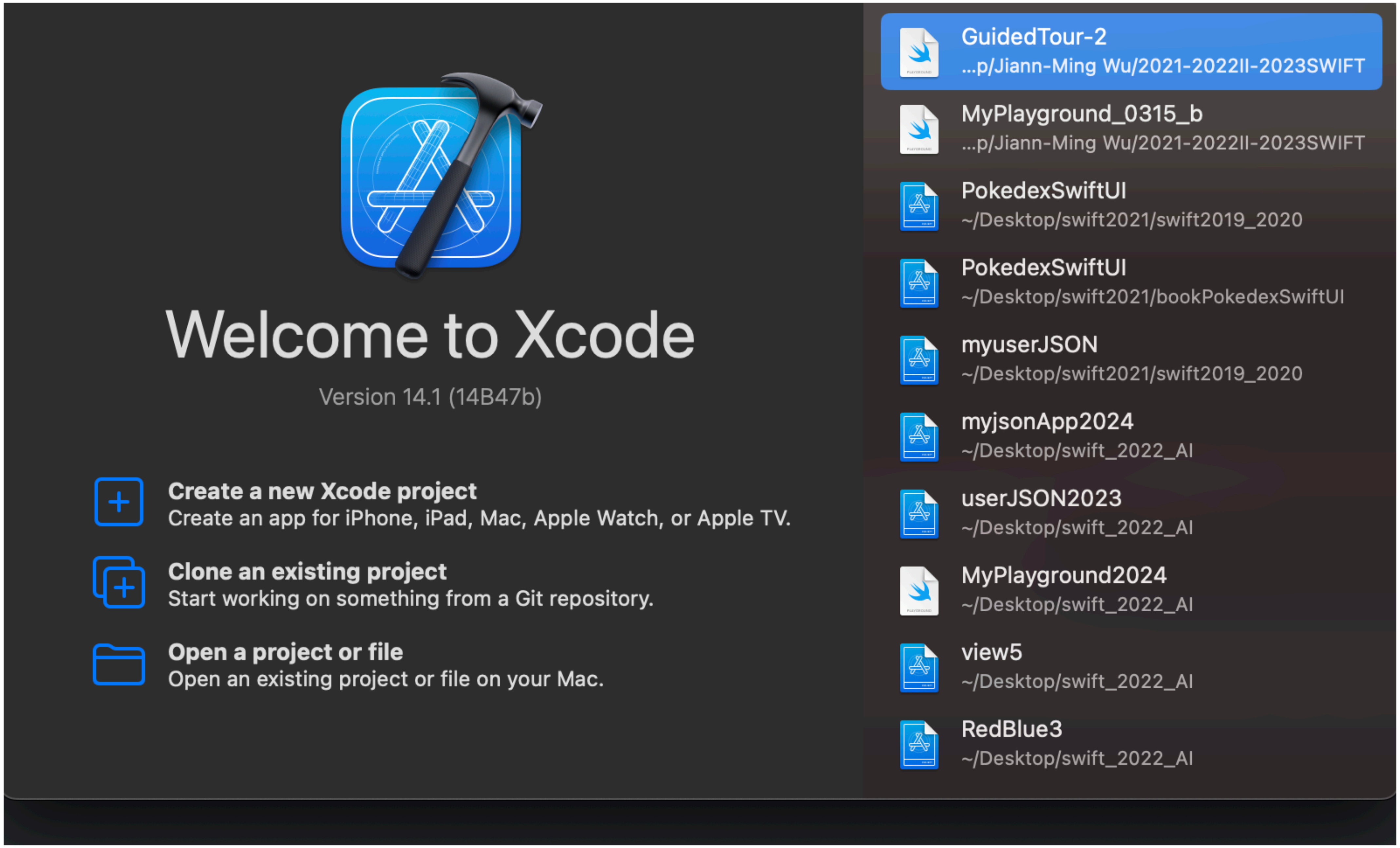
WebViewSwiftUI
~/Desktop/swift_2019_2020

CF temperature
~/Desktop/swift_2019_2020

KMeansSwift
...op/swift_2019_2020/KMeans-Swift-master

C2F
~/Desktop/swift_2019_2020

DrawCirclesTutorial
~/Desktop/swift_2019_2020



Welcome to Xcode

Version 14.1 (14B47b)

Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.

Clone an existing project
Start working on something from a Git repository.

Open a project or file
Open an existing project or file on your Mac.

GuidedTour-2
...p/Jiann-Ming Wu/2021-2022II-2023SWIFT

MyPlayground_0315_b
...p/Jiann-Ming Wu/2021-2022II-2023SWIFT

PokedexSwiftUI
~/Desktop/swift2021/swift2019_2020

PokedexSwiftUI
~/Desktop/swift2021/bookPokedexSwiftUI

myuserJSON
~/Desktop/swift2021/swift2019_2020

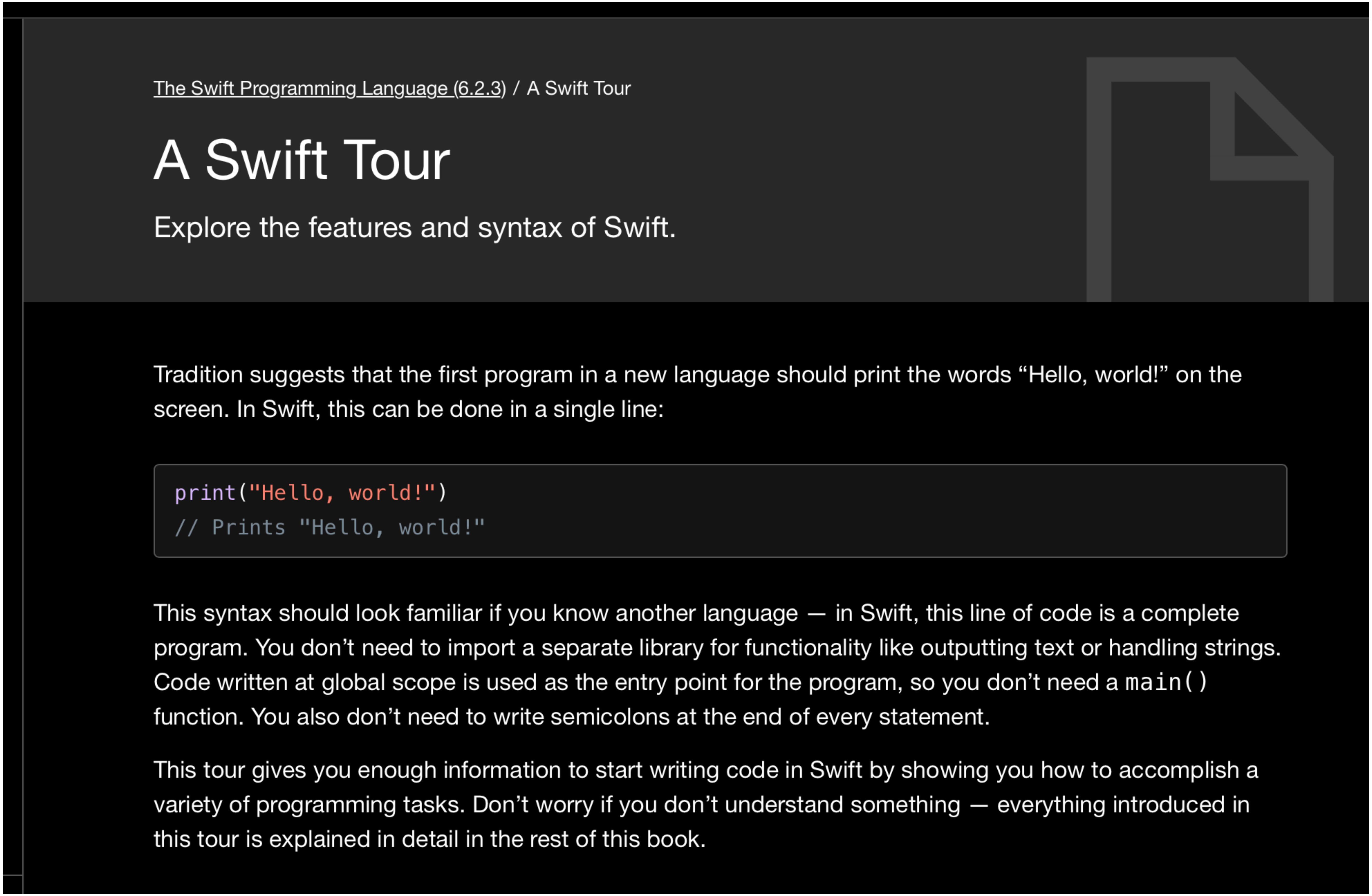
myjsonApp2024
~/Desktop/swift_2022_AI

userJSON2023
~/Desktop/swift_2022_AI

MyPlayground2024
~/Desktop/swift_2022_AI

view5
~/Desktop/swift_2022_AI

RedBlue3
~/Desktop/swift_2022_AI



The Swift Programming Language (6.2.3) / A Swift Tour

A Swift Tour

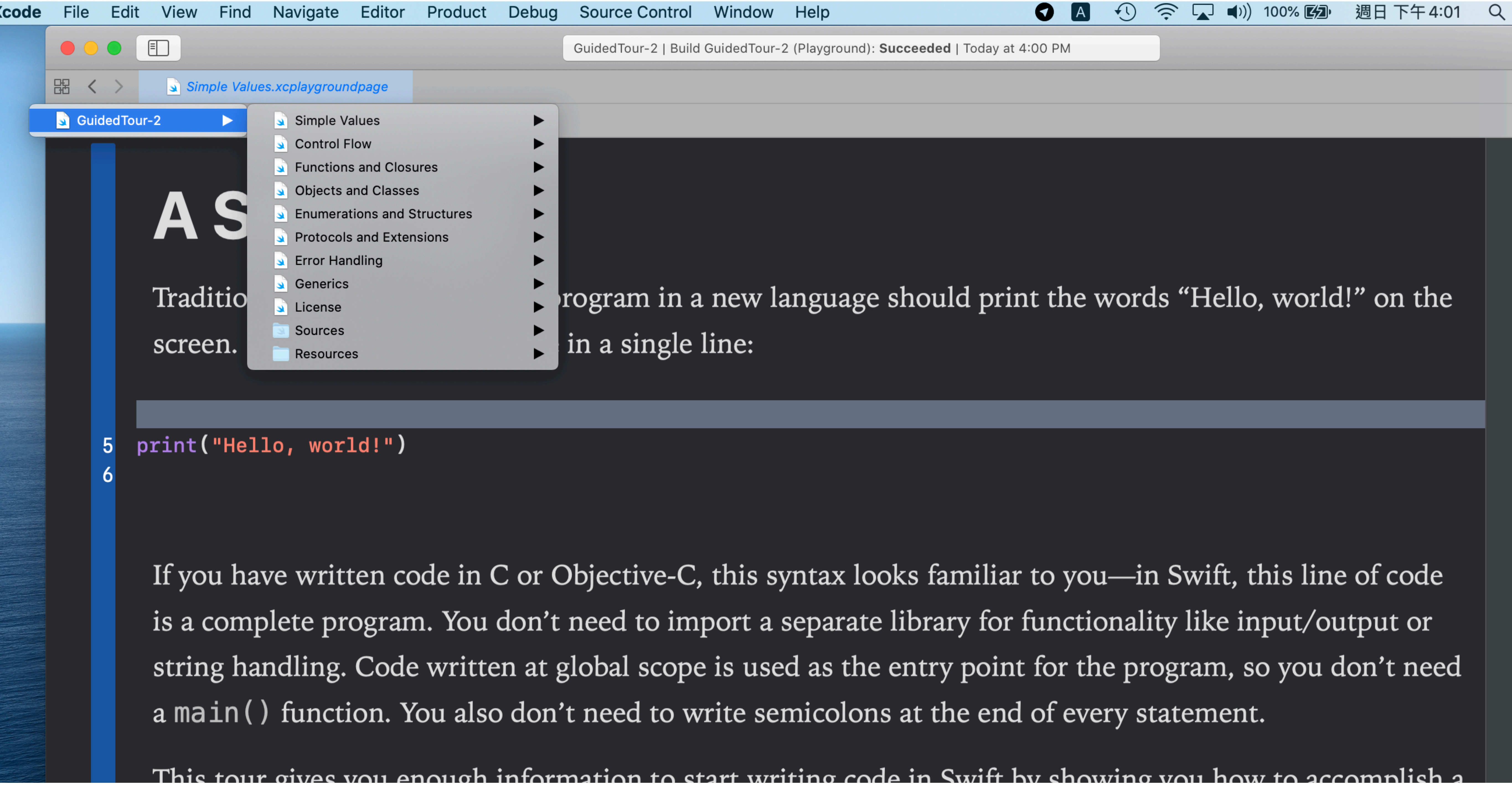
Explore the features and syntax of Swift.

Tradition suggests that the first program in a new language should print the words “Hello, world!” on the screen. In Swift, this can be done in a single line:

```
print("Hello, world!")
// Prints "Hello, world!"
```

This syntax should look familiar if you know another language — in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like outputting text or handling strings. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a variety of programming tasks. Don’t worry if you don’t understand something — everything introduced in this tour is explained in detail in the rest of this book.



GuidedTour-2 | Build GuidedTour-2 (Playground): **Succeeded** | Today at 4:00 PM

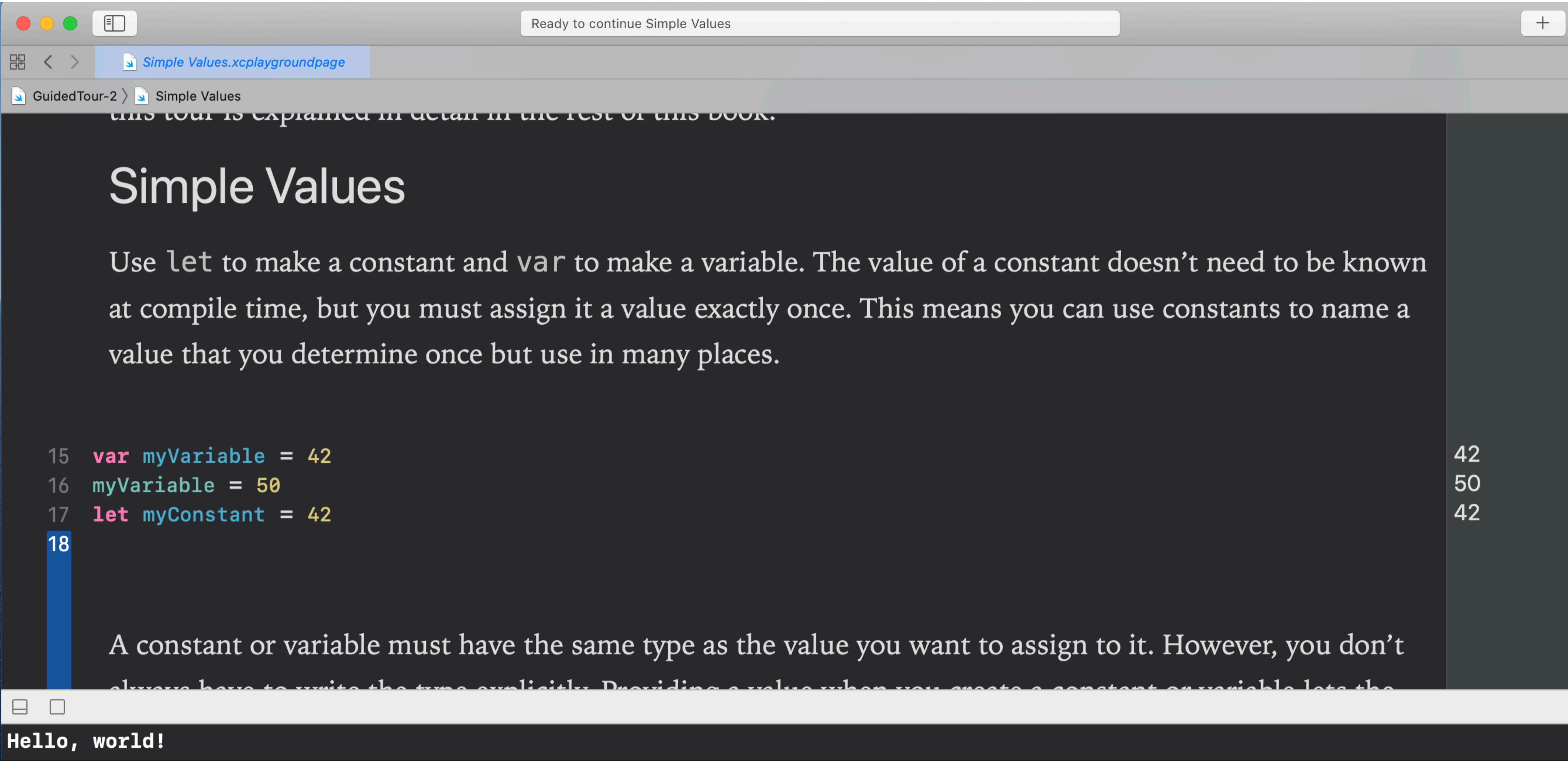
A S

Traditio rogram in a new language should print the words “Hello, world!” on the screen. in a single line:

```
print("Hello, world!")
```

If you have written code in C or Objective-C, this syntax looks familiar to you—in Swift, this line of code is a complete program. You don’t need to import a separate library for functionality like input/output or string handling. Code written at global scope is used as the entry point for the program, so you don’t need a `main()` function. You also don’t need to write semicolons at the end of every statement.

This tour gives you enough information to start writing code in Swift by showing you how to accomplish a



Ready to continue Simple Values
Simple Values.xcplaygroundpage
GuidedTour-2 〉 Simple Values

this tour is explained in detail in the rest of this book.

Simple Values

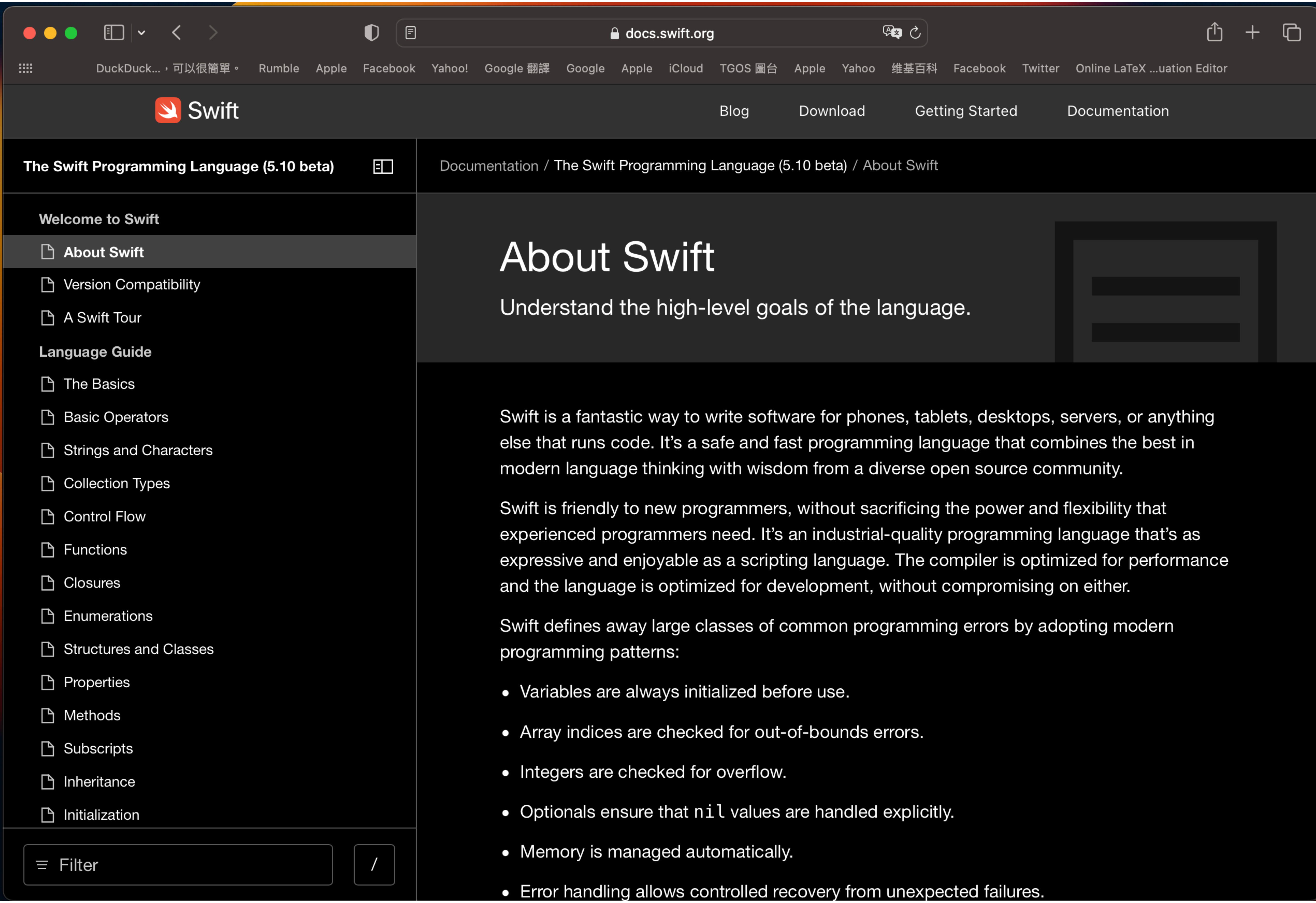
Use `let` to make a constant and `var` to make a variable. The value of a constant doesn’t need to be known at compile time, but you must assign it a value exactly once. This means you can use constants to name a value that you determine once but use in many places.

```
var myVariable = 42
myVariable = 50
let myConstant = 42
```

```
42
50
42
```

A constant or variable must have the same type as the value you want to assign to it. However, you don’t always have to write the type explicitly. Providing a value when you create a constant or variable lets the

```
Hello, world!
```



About Swift

Understand the high-level goals of the language.

Swift is a fantastic way to write software for phones, tablets, desktops, servers, or anything else that runs code. It’s a safe and fast programming language that combines the best in modern language thinking with wisdom from a diverse open source community.

Swift is friendly to new programmers, without sacrificing the power and flexibility that experienced programmers need. It’s an industrial-quality programming language that’s as expressive and enjoyable as a scripting language. The compiler is optimized for performance and the language is optimized for development, without compromising on either.

Swift defines away large classes of common programming errors by adopting modern programming patterns:

- Variables are always initialized before use.
- Array indices are checked for out-of-bounds errors.
- Integers are checked for overflow.
- Optionals ensure that `nil` values are handled explicitly.
- Memory is managed automatically.
- Error handling allows controlled recovery from unexpected failures.