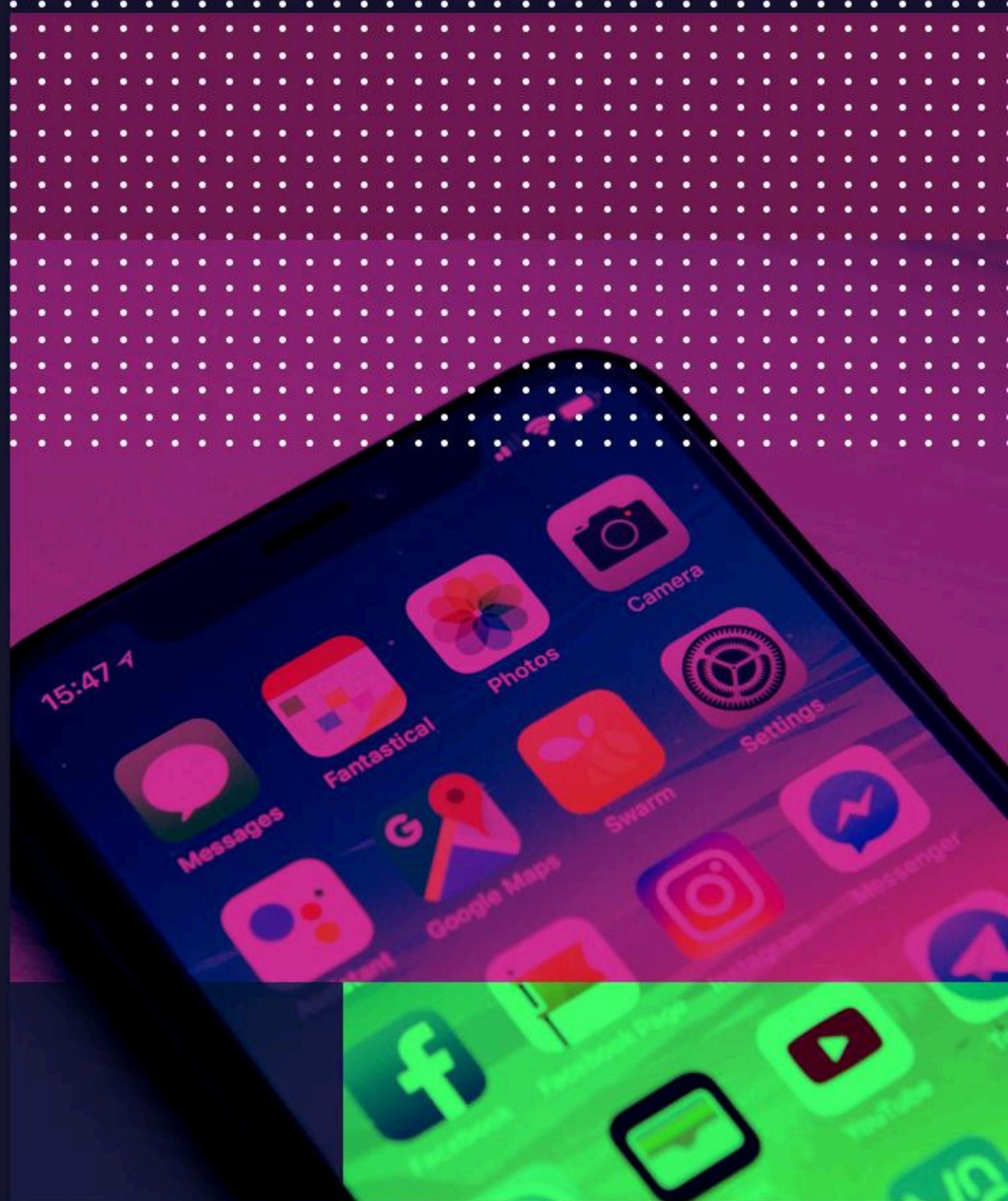


WORKSHOP

How to build an iOS app completely from scratch



- Co-founder, Swift developer at Qminder
- Speaker and mentor
- Open source and inclusivity

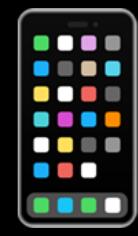


Mr. Byte



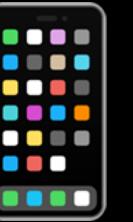
Plan

What is iOS development?



Lunch 

Build and app →



January 9, 2007



2.2 billion

2.2 million

120\$ billion



Mac



iPhone



Apple Watch



iPad



Apple TV

Swift

June 2, 2014 / December 3, 2015

Developed by Apple

General purpose language

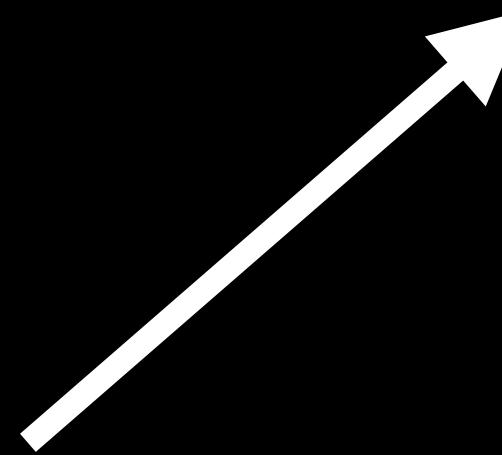


*“A programming language is a vocabulary
and set of grammatical rules for instructing a
computer or computing device to perform
specific tasks.”*

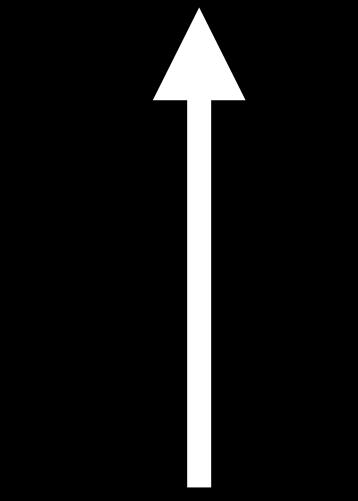
Programming ABC

Variables

```
var x: Int = 1
```



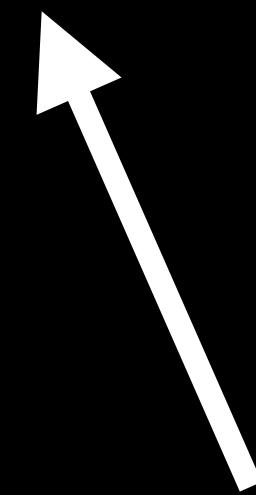
keyword



name



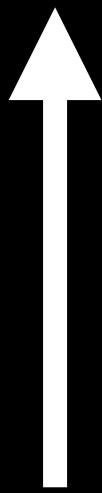
data type



value

Constants

```
let pi: Double = 3.14
```



keyword

Different variables

```
var number: Int = 13
```

```
var decimal: Float = 2.76
```

```
var name: String = "RigaTechGirls 🧑💻"
```

```
var isItSpringNow: Bool = false
```

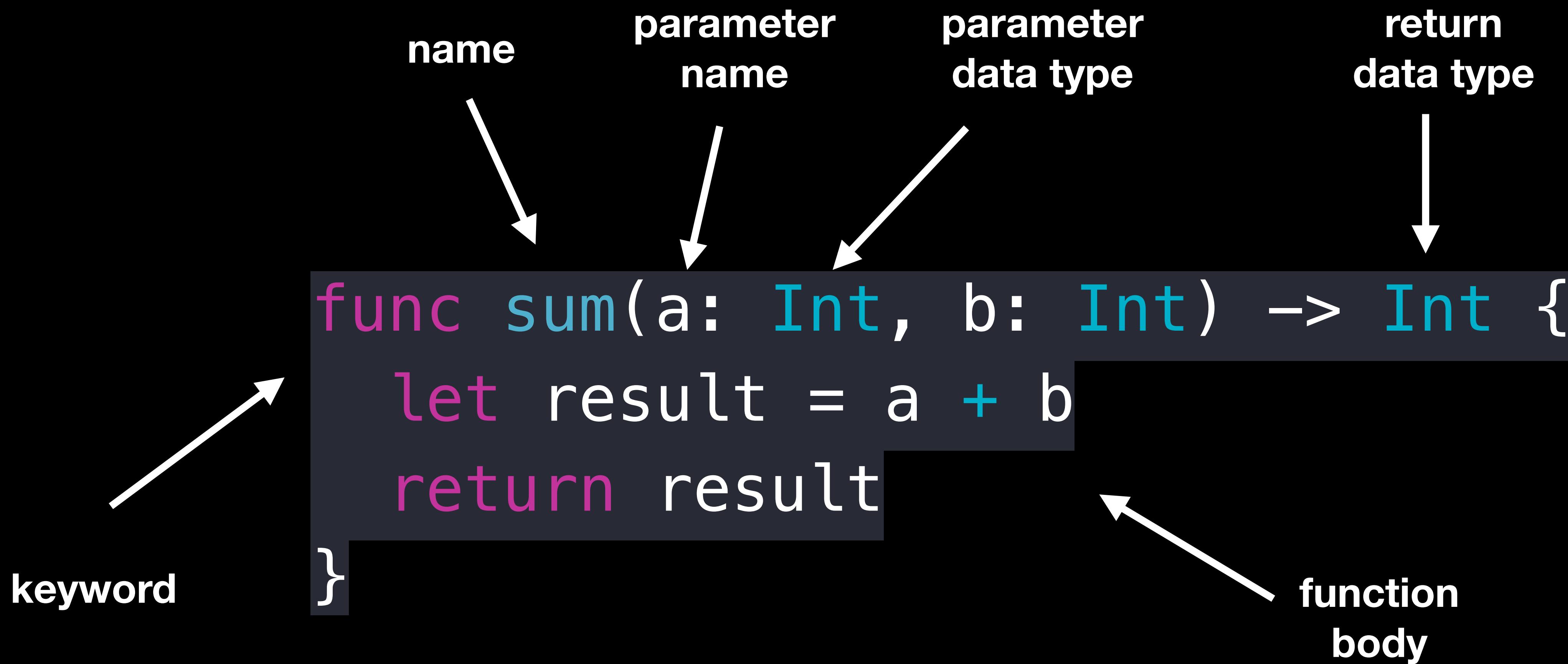
Optional variables



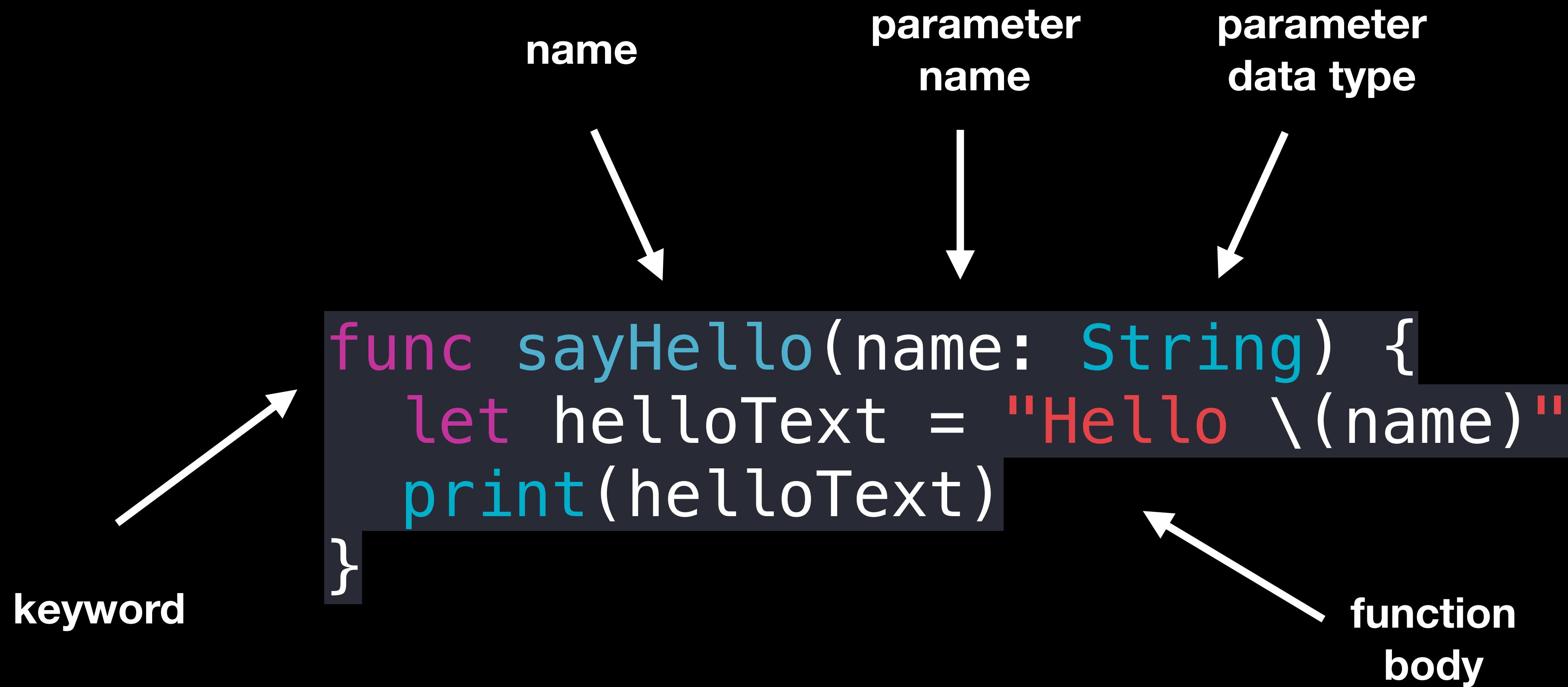
```
var myOptional: Int? = 4
```

```
var myOptional: Int? = nil
```

Functions



Functions



Conditionals

```
if day == "Monday" {  
    // work hard  
}  
else if day == "Friday" {  
    // play hard  
}  
else {  
    // rest  
}
```

condition

keyword

The diagram illustrates the structure of a conditional statement. It features a large dark gray rectangular box containing the code. Four white arrows originate from labels on the left and point to specific tokens within the code:

- An arrow labeled "condition" points to the string "Monday" in the first branch of the if-else-if ladder.
- An arrow labeled "keyword" points to the opening brace "{" of the outermost if block.
- An arrow labeled "keyword" points to the opening brace "{" of the else-if block.
- An arrow labeled "keyword" points to the opening brace "{" of the else block.

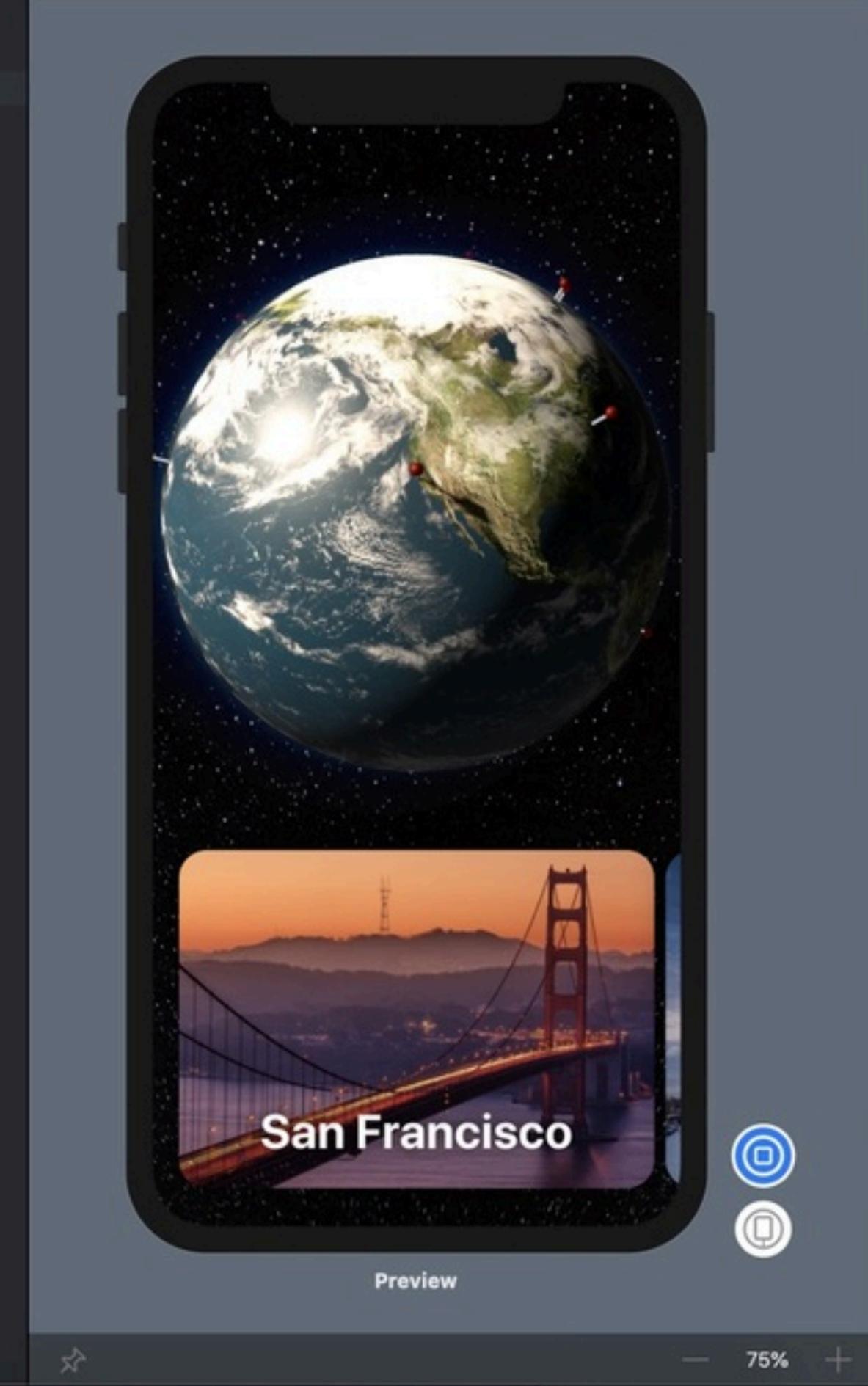
Tools

Xcode

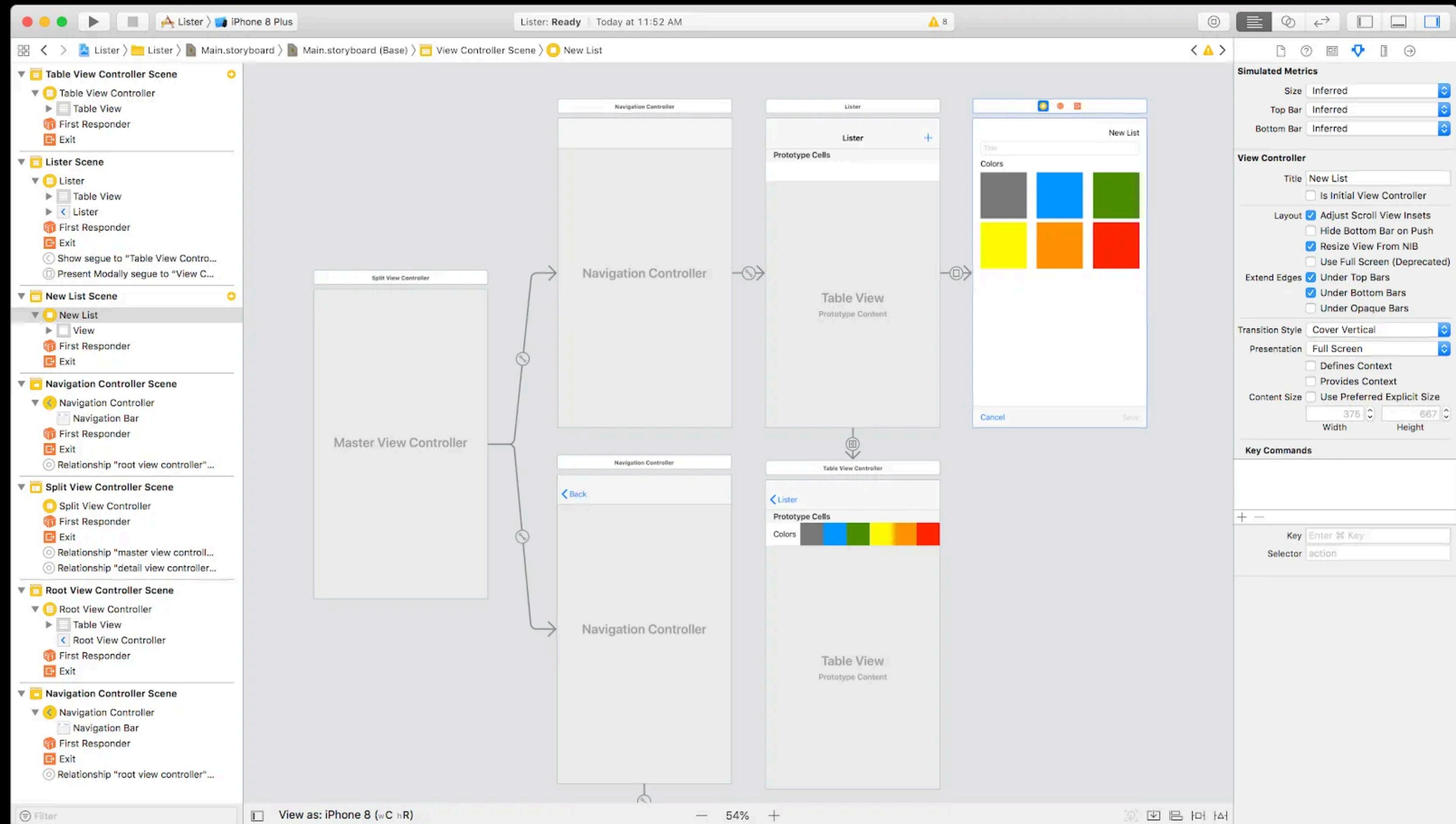
The screenshot shows the Xcode interface with a project named "Travel". The left sidebar displays the file structure:

- Travel (root folder)
 - ReadMe.md
 - Travel (folder)
 - Travel.entitlements
 - AppDelegate.swift
 - Model (folder)
 - Discover (folder)
 - DiscoverView.swift
 - DiscoverTileView.swift
 - GlobeView.swift
 - DestinationsListView.swift
 - PagingScrollView.swift
 - CardsController.swift
 - Globe Scene (folder)
 - Plan (folder)
 - Journal (folder)
 - Journal.storyboard
 - StoryboardHostView.swift
 - JournalView.swift
 - JournalViewController.swift
 - JournalViewCell.swift
 - JournalTableController.swift
 - JournalDetailTableViewCell.swift
 - JournalPreviewViewCell.swift
 - JournalHeaderView.swift
 - JournalAddController.swift
 - JournalImagePickerController.swift
 - RoundedCornerButton.swift
 - Weather (folder)
 - Login Screen (folder)
 - Login.storyboard
 - LoginViewController.swift
 - ForgotPasswordController.swift
 - ForgotPasswordController.xib
 - ForgotPasswordStatusView.swift
 - Main Screen (folder)

```
8 import SwiftUI
9
10 struct DiscoverView : View {
11     let sceneController: GlobeSceneController
12     @State private var selection: Region? = nil
13
14     var body: some View {
15         let pagingScrollViewController =
16             sceneController.pagingScrollViewController
17         pagingScrollViewController.didChangeToPageHandler = { page in
18             self.selection = DataSource.shared.regions[page]
19         }
20
21         return GeometryReader { container in
22             return ZStack(alignment: .bottom) {
23                 GlobeView(
24                     selection: self.$selection,
25                     sceneController: self.sceneController
26                 )
27
28                 PagingTilesView(
29                     containerSize: container.size,
30                     pagingScrollViewController: pagingScrollViewController
31                 ) { region in
32                     self.selection = region
33                 }
34             }
35             .background(Color.black)
36         }
37     }
38
39 struct PagingTilesView<T> : View where T : PagingScrollViewController {
40     let containerSize: CGSize
41     let pagingScrollViewController: T
42     var selectedTileAction: (Region) -> ()
43
44     var body: some View {
45         let tileSize = containerSize.width * 0.9
46         let tileHeight = CGFloat(240.0)
47         let verticalTileSpacing = CGFloat(8.0)
```



User Interface



iOS Simulator

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure under "Solar System".
- Editor:** Displays the code for `MainViewController.swift`. The code initializes a `SceneViewController` and adds its view as a subview of the content container view.
- Preview:** An iPhone X simulator window shows the app's interface. The title bar says "Running Solar System iOS on iPhone X". The main screen is titled "Details" with a back button labeled "Planets".
- Right Panel:** Shows the "Planets" screen with cards for Mercury, Venus, Earth, Mars, and Jupiter, each with a circular image and a brief description.

Let's build an app



Step 1 - Create project

New Xcode project

Select iOS Simulator

Run empty app

Step 2 - Assets

Download assets

Add to Asset Catalog

Dark mode

Step 3 - User Interface

Text labels

Inputs

Flags

Buttons

Step 4 - Layout elements

Put elements on the screen

Constraints

Screen resolutions

Step 5 - Programming

Convert function from IDR to EUR

Format number

Step 6 - Connect UI

IBOutlets

IBActions

Connect interface to code

Step 7 - Advanced I

Convert automatically

Step 8 - Advanced II

Convert vice versa

Step 9 - Extra

Build currency selection

VDN has more crazy rate

What's next?

Treehouse

Hacking with Swift

Just start building



fassko

www.kristaps.me

