

iOS Development

By
S.Abhinav

1. What iOS is.

2. Why iOS is powerful.

3. Becoming a developer.

What is iOS?

iOS -> Operating System used in Apple Devices



Hierarchy of iOS.

Cocoa Touch - UI Elements

Media / Application Services

Core Services

Core OS / iOS kernel

Why is iOS powerful?

iOS SDK

The iOS SDK gives you tools for building great apps.

UI Development - Multi-touch events and control, Accelerometer support, View hierarchy

Graphics - OpenAL, Video playback, Image file formats, Core Animation, OpenGL ES

Core Services - Networking, Embedded SQLite database
Core Location, Threads

Mac OS X Kernel - TCP/IP , Sockets , Power management, File system, Security

Vertical Integration

All apps are built specifically for Apple hardware

All aspects of user experience is regulated by Apple

Advantages

Less security issues

Privacy

Smoothness in User Experience

Latest Frameworks

Apple constantly releases cutting edge frameworks

MapKit

EventKit

Messages

CoreML

Apple Pay

CloudKit

ARKit

GameKit

HealthKit

Latest Frameworks



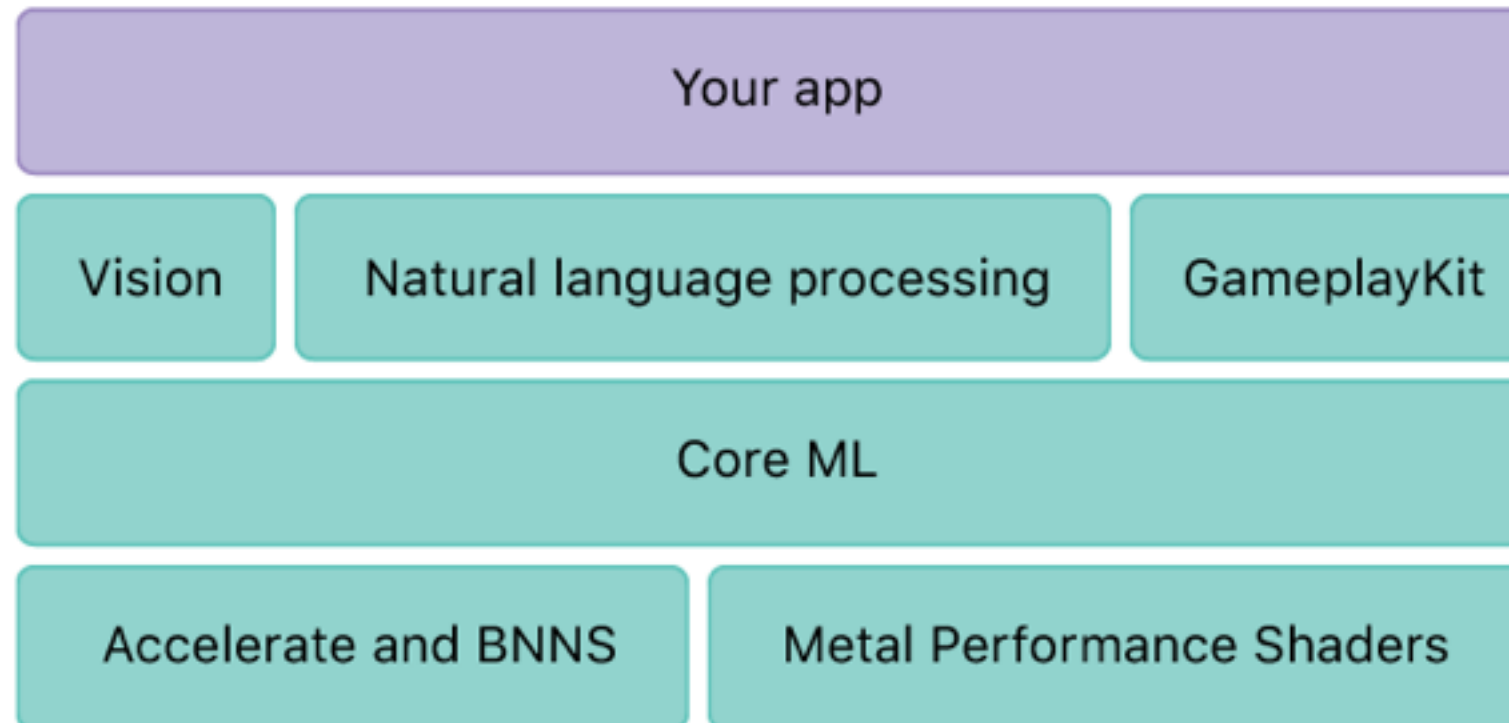
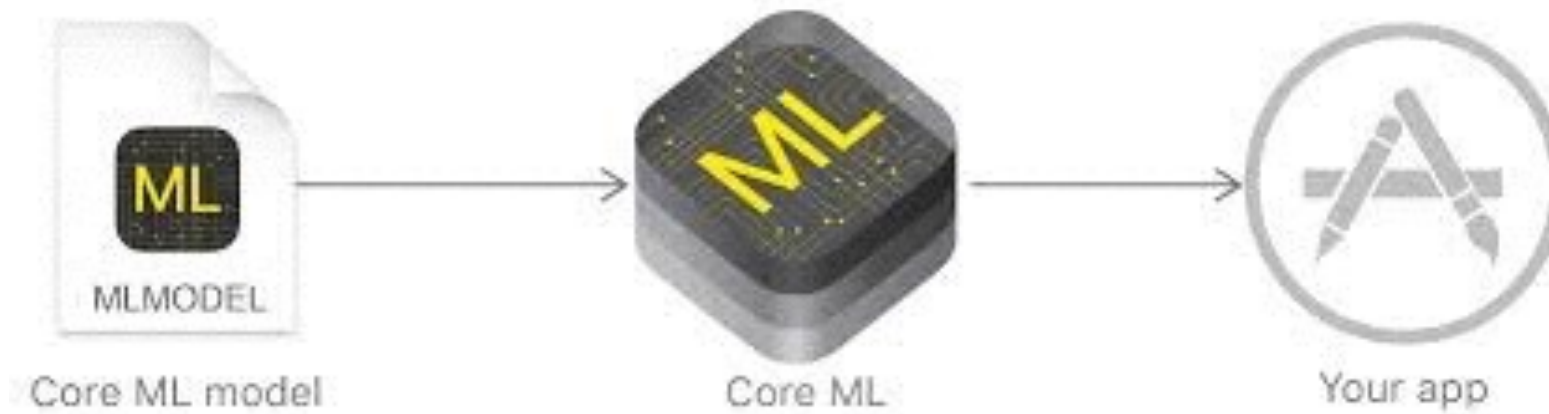
ARKit



HealthKit



Latest Frameworks



Developer Community



Forums



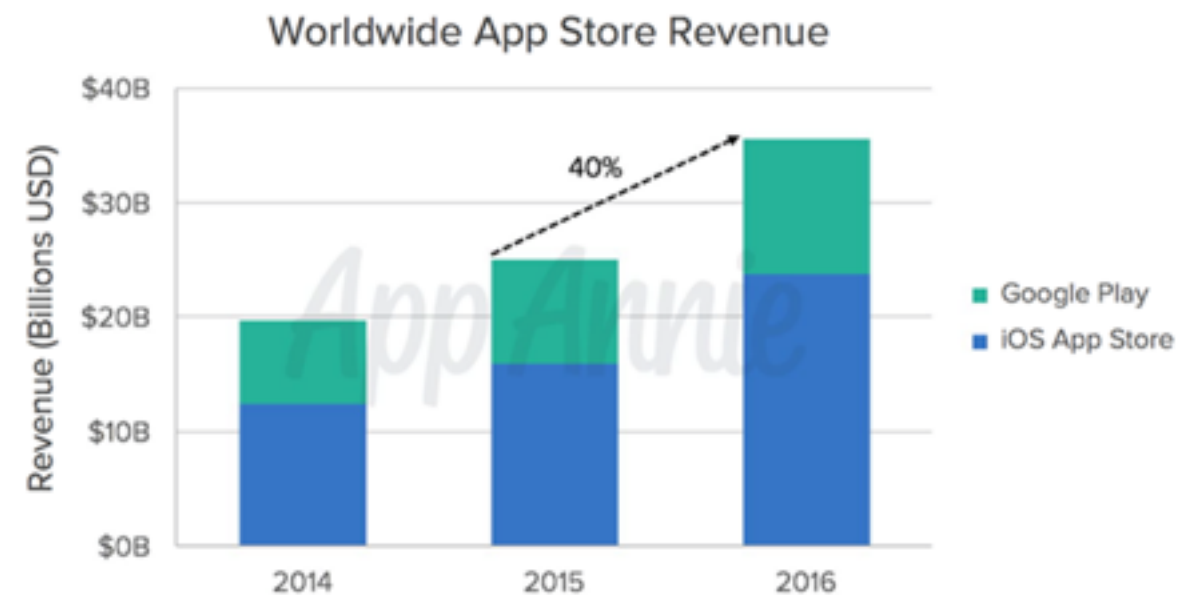
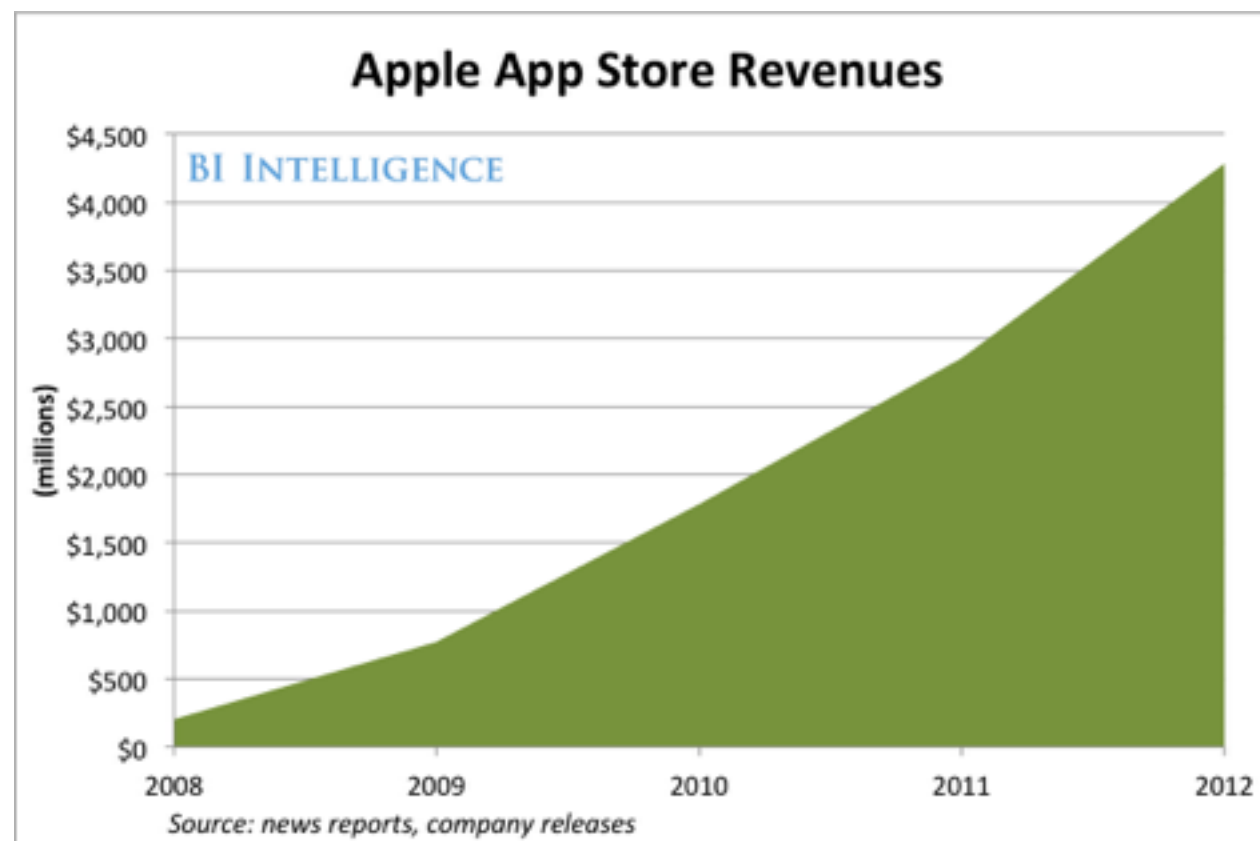
Conferences



Bug Reporter

Mass adoption

iOS is the second most popular mobile operating system in the world, after Android.



Becoming a developer

Important Terminologies

Xcode - The IDE used for iOS development

Swift - The language used inside Xcode

iOS - The platform for development

We use Swift inside Xcode to develop for iOS

Swift



Powerful and intuitive programming language

Concise (let count =5)

Safe

References

Online portals - Udemy/Udacity/Coursera

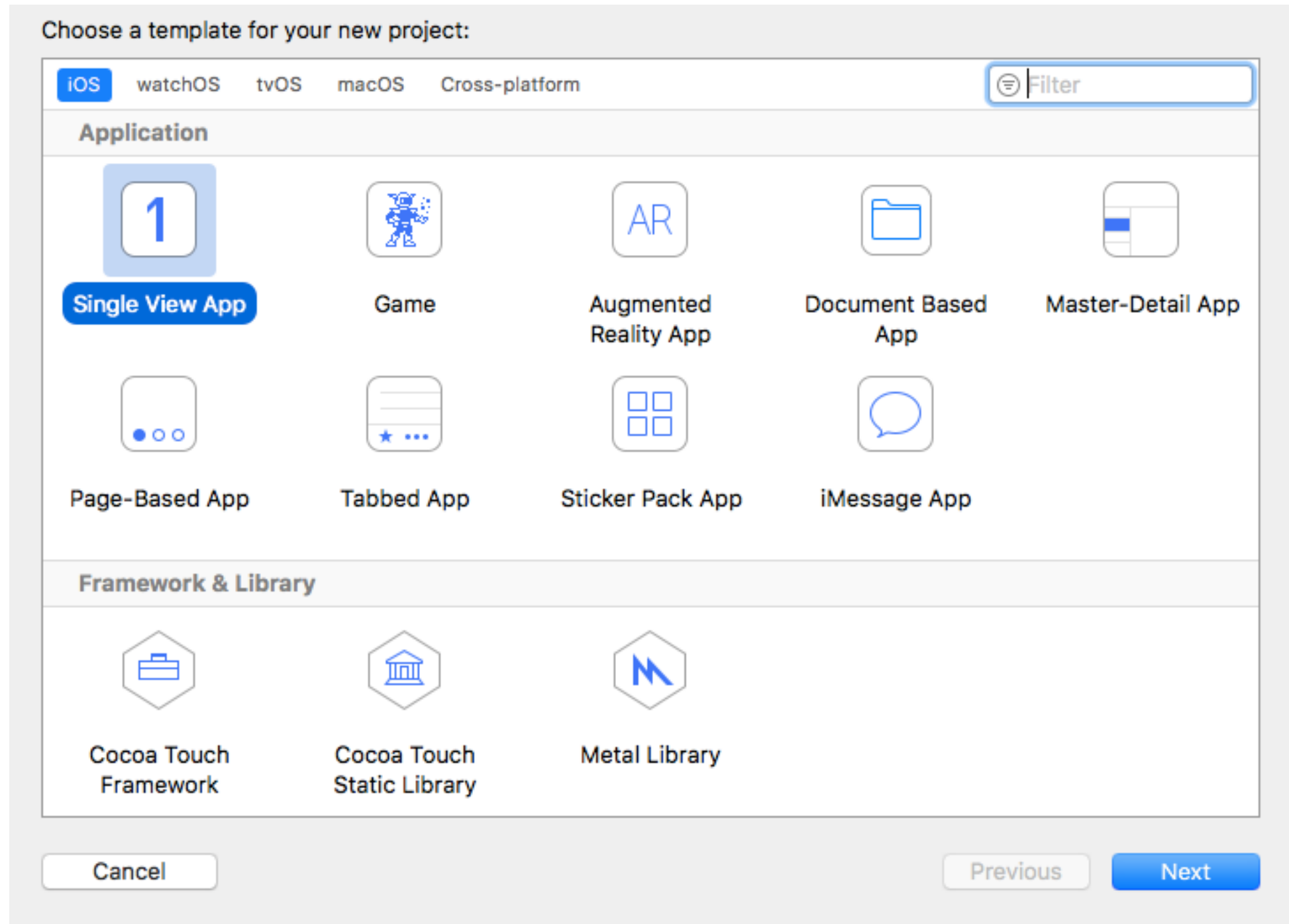
Apple Documentations

Apple Developer Website

Online Blogs - Ray Wenderlich iOS tutorials

Code

Create a new single view app.



Code


Select from a list of UI Components


Label **Label** - A variably sized amount of static text.

Button **Button** - Intercepts touch events and sends an action message to a target object when it's tapped.

1 2 Segmented Control - Displays multiple segments, each of which functions as a discrete button.


Text **Text Field** - Displays editable text and sends an action message to a target object when Return is tapped.

 **Slider** - Displays a continuous range of values and allows the selection of a single value.

 **Switch** - Displays an element showing the boolean state of a value. Allows tapping the control to toggle the value.

 **Activity Indicator View** - Provides feedback on the progress of a task or process of unknown duration.

 **Progress View** - Depicts the progress of a task over time.

 **Page Control** - Displays a dot for each open page in an application and supports sequential navigation through the pages.

Code

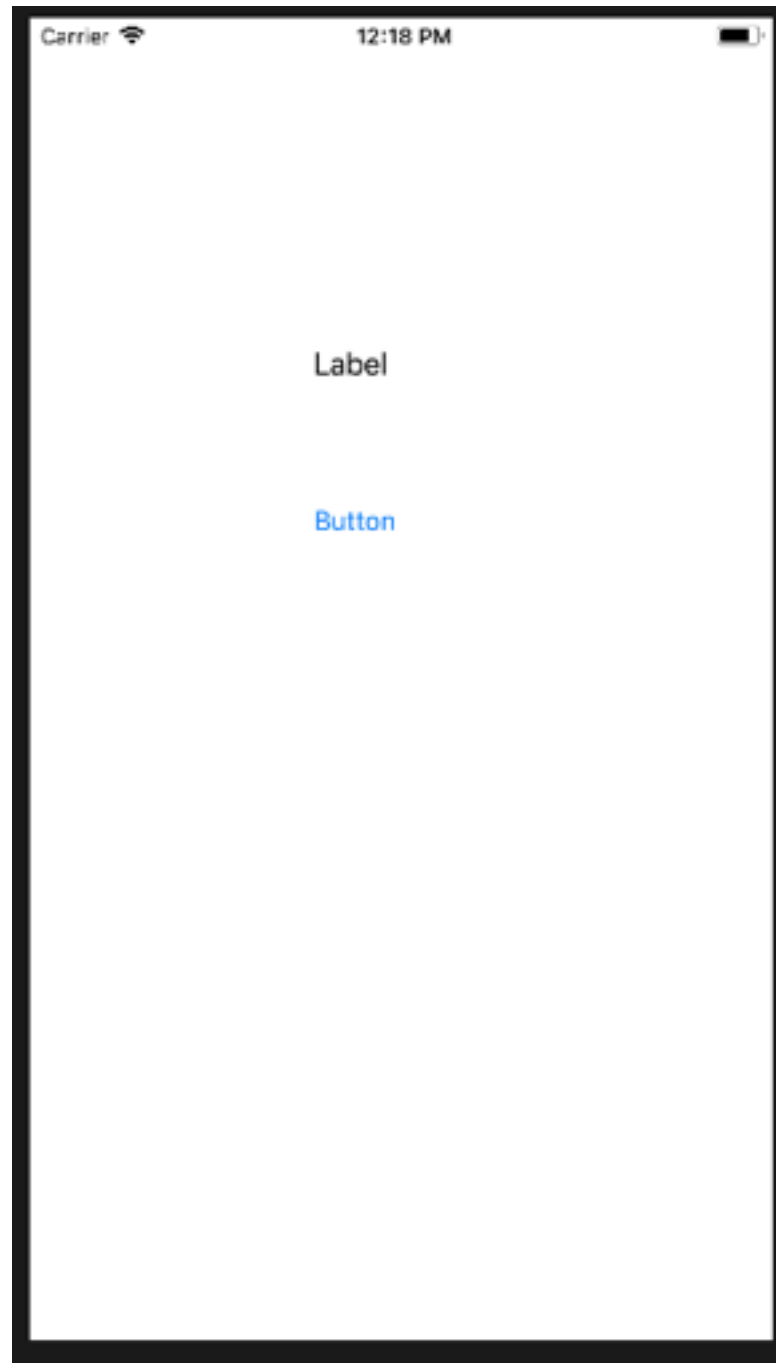
Drag and drop the required functionality

Label

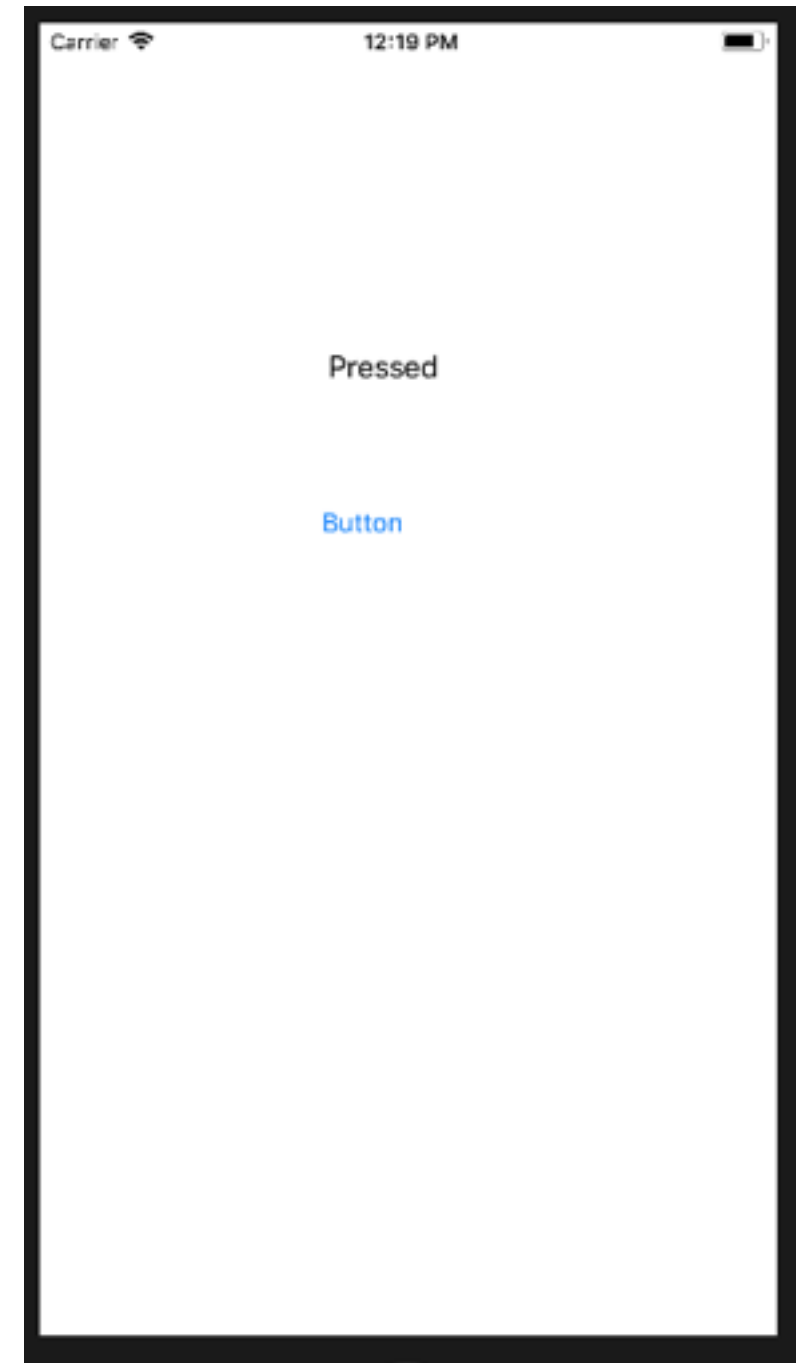


```
6 // Copyright © 2018 TenPal. All
   // rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController:
    UIViewController {
12
13     @IBAction func Button1(_
        sender: Any) {
14
15         Label1.text = "Pressed";
16
17     }
18
19     @IBOutlet weak var Label1:
        UILabel!
20 }
```

Code



Press Button
→



Code

Maps can be initialised in a similar way

```
locationManager.delegate = self
locationManager.desiredAccuracy = kCLLocationAccuracyBest
locationManager.requestAlwaysAuthorization()
locationManager.startUpdatingLocation()

firstMap.addGestureRecognizer(panGestureRecognizer)

//initialize latitude and longitude
let latitude = 12.99
let longitude = 80.23

//get center value
center = CLLocationCoordinate2DMake(latitude,longitude)
let latitudeDelta = CLLocationDegrees(0.05)

let longitudeDelta = CLLocationDegrees(0.05)

let span = MKCoordinateSpan(latitudeDelta: latitudeDelta,
                             longitudeDelta: longitudeDelta)

let region = MKCoordinateRegion.init(center: center ,span: span )

firstMap.setRegion(region, animated: false)
```

Questions