

appropos  
academy





# iOS Basics with Swift





# Introduction

- Name
- Experience
- Expectation



# Schedule

	Day 1	Day 2	Day 3
9:00 - 10:30	Introduction + Setup	Storyboards	Networking
10:45 - 12:15	Swift I	Navigation + Tables	Dependencies + Wishlist
13:15 - 14:45	Swift II	Assets + Scheduling	Testing + Platform
15:00 - 16:30	Apps + Xcode	Finishing the Game	Distribution + Feedback



# Let's start!

Please interrupt me if ...

- ... you need a break
- ... you want me to repeat something
- ... you have a question



# iOS

What's different?



# Limitations

- Screen
- Energy
- Mobile Data
- Memory



# Expectations

- Stability
- Responsiveness
- Energy Consumption
- Data Consumption
- User Interface (HIG)

<https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/>





# Swift



# The Language

*After Apple unveiled the Swift programming language, it quickly became one of the **fastest growing** languages in history. Swift makes it easy to write software that is incredibly **fast and safe** by design. Now that Swift is **open source**, you can help make the best general purpose programming language available everywhere.*

*For students, learning Swift has been a great introduction to modern programming concepts and best practices. And because it is now open, their Swift skills will be able to be applied to an even broader range of platforms, from **mobile** devices to the **desktop** to the **cloud**.*

- **The Swift Team** - <https://docs.swift.org>



# Safety

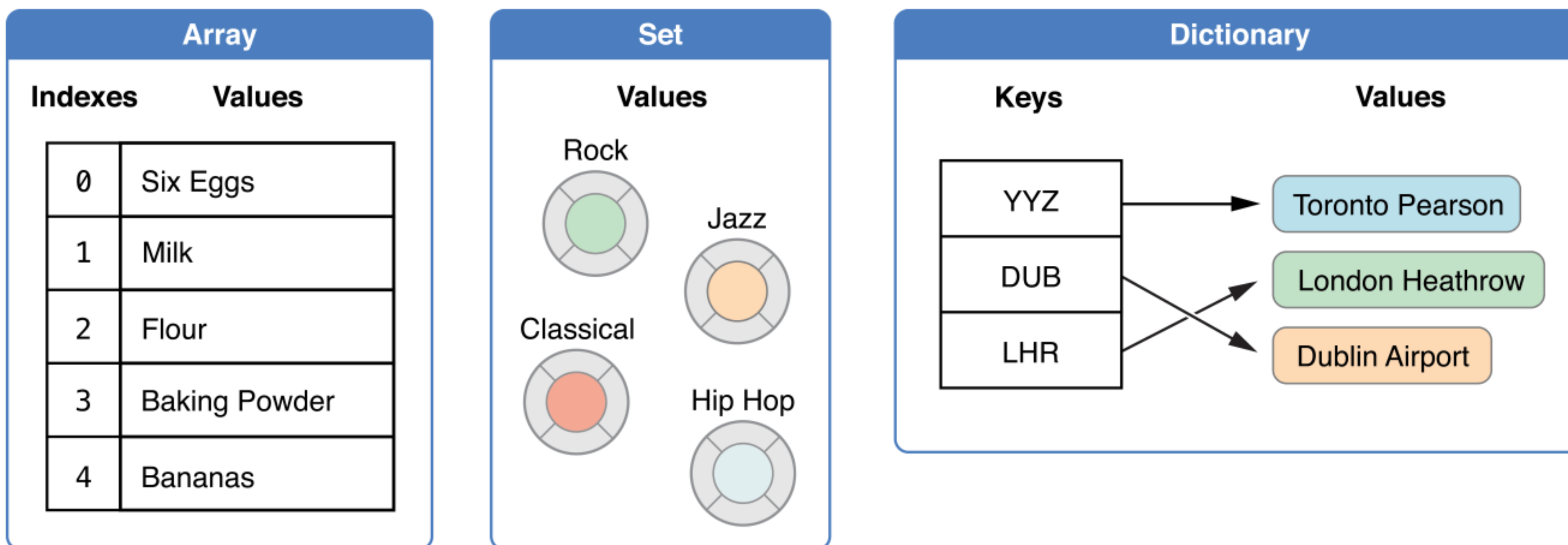
*Swift defines away large classes of common programming errors by adopting modern programming patterns:*

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

- **The Swift Programming Language (Swift 5)** - <https://docs.swift.org/swift-book/>



# Collection Types

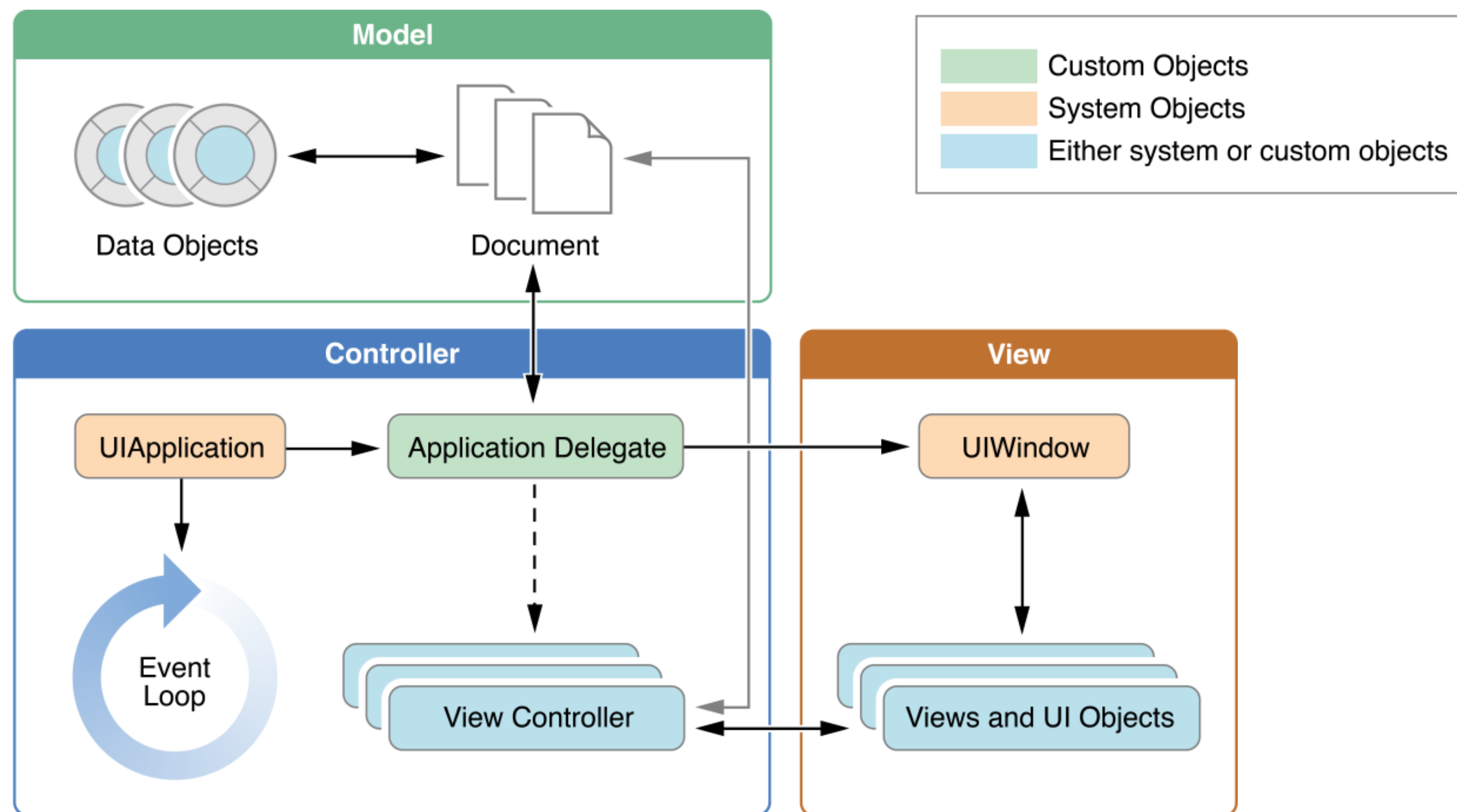




# Apps

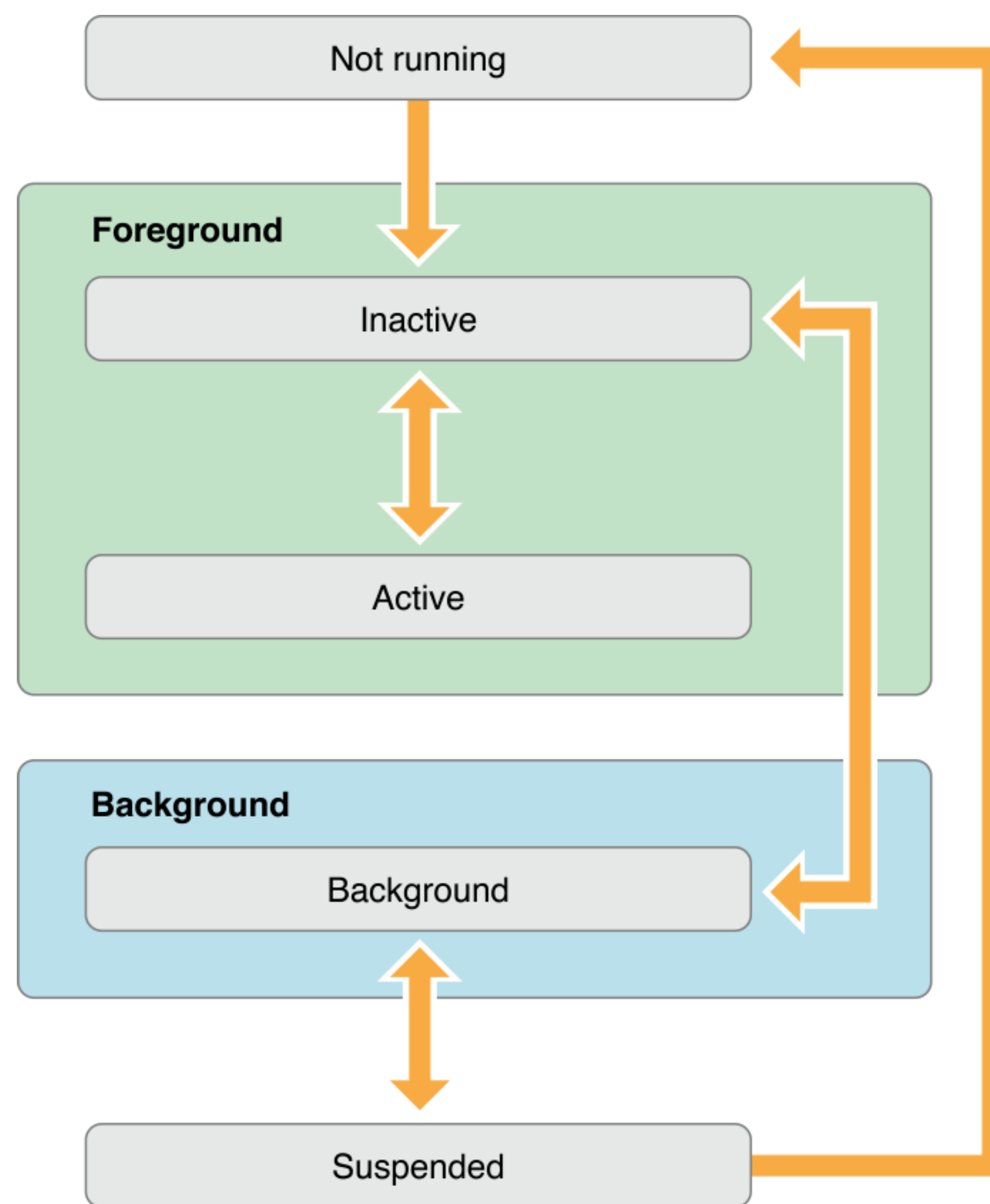


# App Architecture





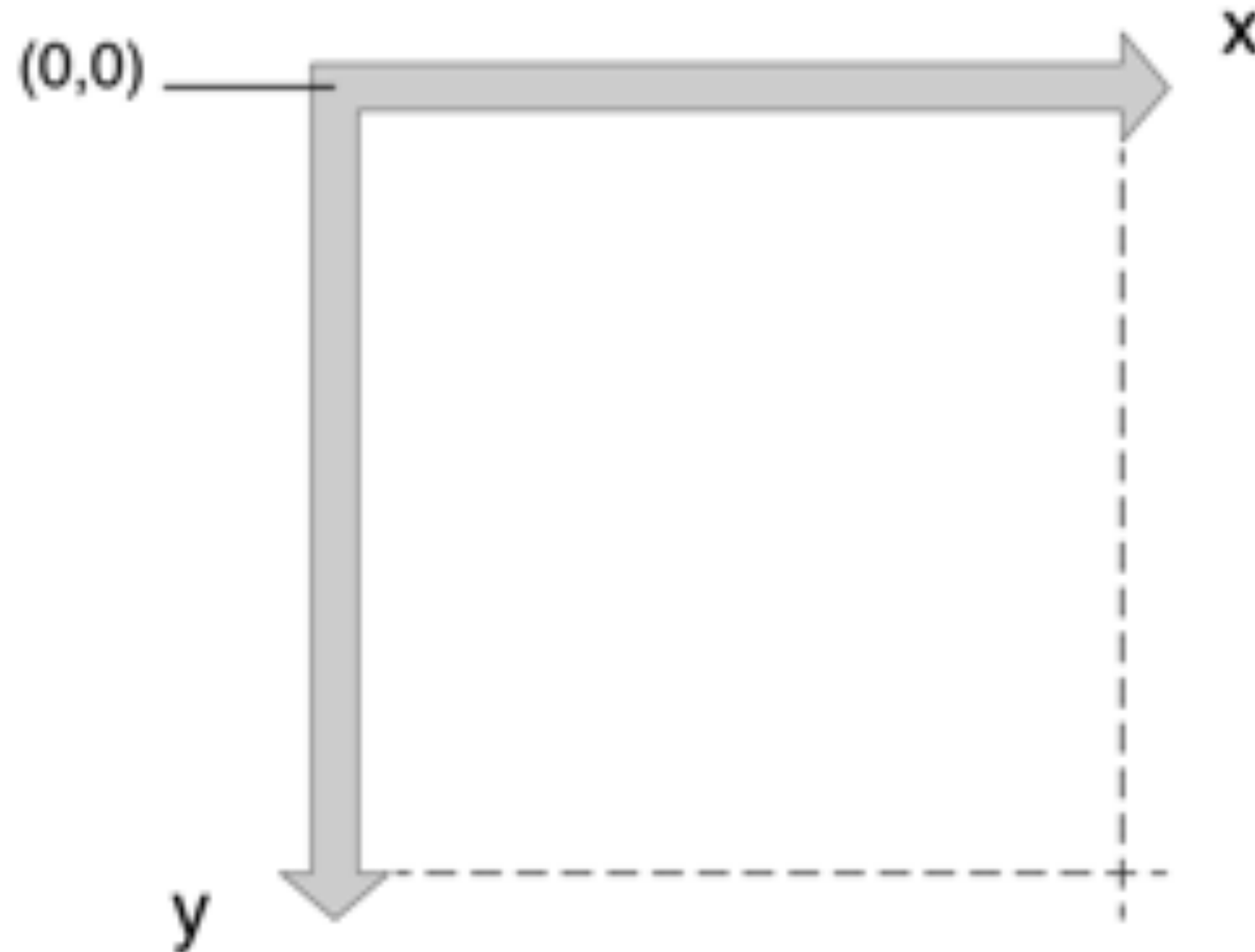
# App Lifecycle







# The UI Coordinate System







# Let's code!

a very simple game



# Task 1

- UIViewController with button and label
- tapping the button appends an “!” to the label text



# Task 2

- UIViewController with 3 **game buttons** and a start button
- start button tapped > **mark** a random **game button**
- **marked** button tapped > **unmark** the button



# Task 3

- UITableViewController with 5+ **cells**
- **cell** gets **marked** on tap (change text)
- **cell** gets **unmarked** on another tap (change text)



# Task 4

- UITableViewController with 5+ **image cells**
- **image cell** gets **marked** on tap (show image)
- **image cell** gets **unmarked** on another tap (hide image)



# Dispatch Queues

FIFO queues to which your application can submit tasks in the form of block objects

- **Main queue:** runs on the main thread and is a serial queue
- **Global queues:** four concurrent queues that are shared by the whole system.  
Priorities: high, default, low, and background (is throttled in any I/O activity to minimize negative system impact)
- **Custom queues:** self-created queues, can be serial or concurrent (blocks in these queues actually end up in one of the global queues)



# Dispatch Queues

Blocks are scheduled synchronously or asynchronously

- **Synchronously:** code waits until that block finishes execution
- **Asynchronously:** code continues executing while the block runs elsewhere

Attempting to synchronously execute a block on the main queue  
results in a **deadlock!**



# Task 5

- taps can only **unmark image cells**
- a random **image cell** becomes **marked** after some time
- repeat until all **image cells** are **marked**

Hint: Use `asyncAfter(...)`





# Task 6

- random **marking** accelerates with every tap
- UIAlertController pops up when all **image cells** are **marked**
- bonus: app icon, launch screen, save longest run (taps)



# Dependency Management

CocoaPods	Carthage	Swift Package Manager
old stable default option deeply integrated complex configuration	newer lightweight missing packages more manual steps repository = package	new official well integrated doesn't support iOS yet

Selected libraries: <https://github.com/matteocrippa/awesome-swift>



# The iOS Platform



# Hardware

Connectivity	Sensors	Other
<ul style="list-style-type: none"><li>• GSM/HSPA/LTE</li><li>• WLAN</li><li>• Bluetooth (BLE)</li><li>• GPS</li><li>• NFC*</li></ul>	<ul style="list-style-type: none"><li>• Barometer</li><li>• Three-axis gyro</li><li>• Accelerometer</li><li>• Proximity sensor</li><li>• Ambient light sensor</li><li>• Fingerprint*</li></ul>	<ul style="list-style-type: none"><li>• Camera</li><li>• Speakers</li><li>• 3D Touch</li></ul>



# Software

Content	Graphics	Data
<ul style="list-style-type: none"><li>• WebKit</li><li>• MapKit</li><li>• ARKit</li><li>• iAd</li></ul>	<ul style="list-style-type: none"><li>• Metal</li><li>• OpenGL</li><li>• SceneKit</li><li>• SpriteKit</li></ul>	<ul style="list-style-type: none"><li>• CloudKit</li><li>• EventKit</li><li>• HealthKit</li><li>• Core Data</li></ul>
Devices	Other	
<ul style="list-style-type: none"><li>• CoreBluetooth</li><li>• iBeacon</li><li>• WatchKit</li><li>• HomeKit</li></ul>	<ul style="list-style-type: none"><li>• SiriKit</li><li>• PassKit</li><li>• Game Center</li><li>• Core ML</li></ul>	<ul style="list-style-type: none"><li>• Core Motion</li><li>• UserNotifications</li><li>• AirPlay</li><li>• StoreKit</li></ul>



# Distribution



# Distribution Process

- Enroll into Apple Developer Program
- Create App Store listing (Bundle ID, Texts, Images, ...)
- Submit release build
- Review by Apple (automated + human checks, up to 5 days)
- Approval, rejection or request for changes
- Publishing (up to 1 day until visible everywhere)



# Pricing

## **Apple Developer Program**

Apple ID (free) / Individual (99\$/y) / Organization (99\$/y) / Enterprise (299\$/y)

<https://developer.apple.com/support/compare-memberships/>

## **AppStore**

Free / Freemium (70%) / Paid (70%) / Subscription (70-85%)

<https://developer.apple.com/app-store/business-models/>





# Feedback

- Exercises
- Content
- Knowledge



# App Architecture

