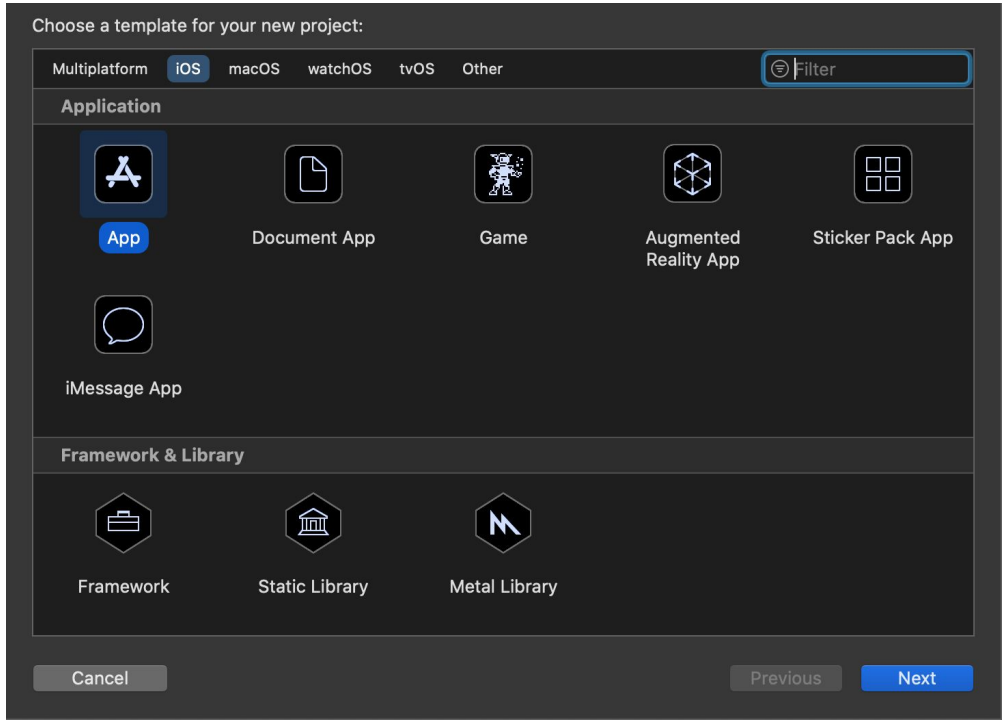
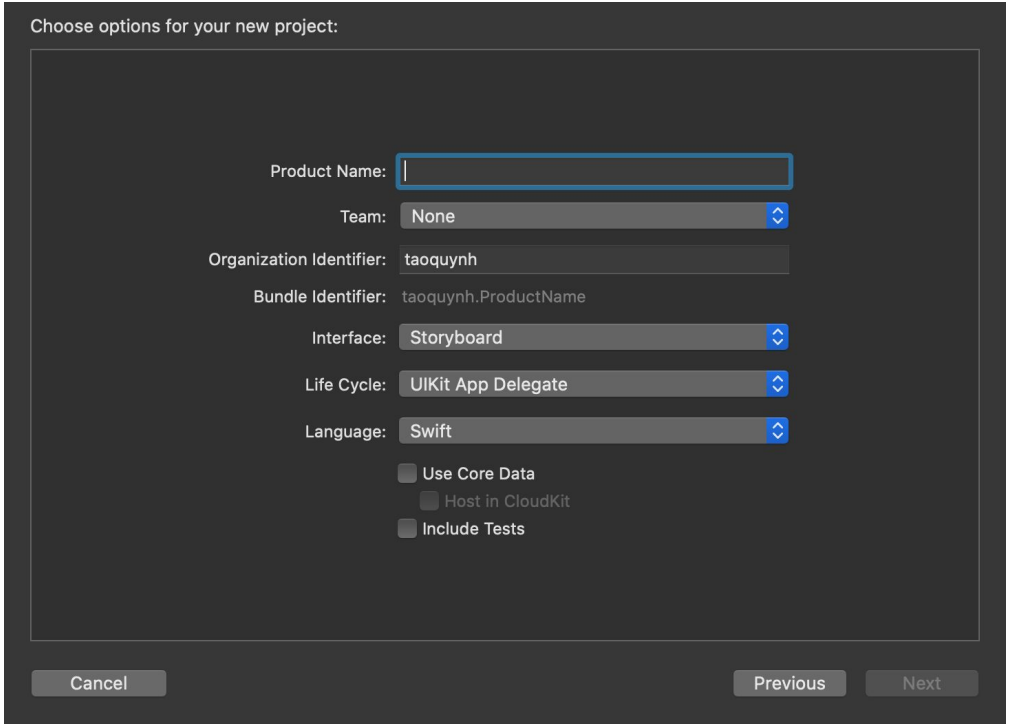
Cấu trúc thư mục iOS Swift

quynh@techmaster.vn

Template Project

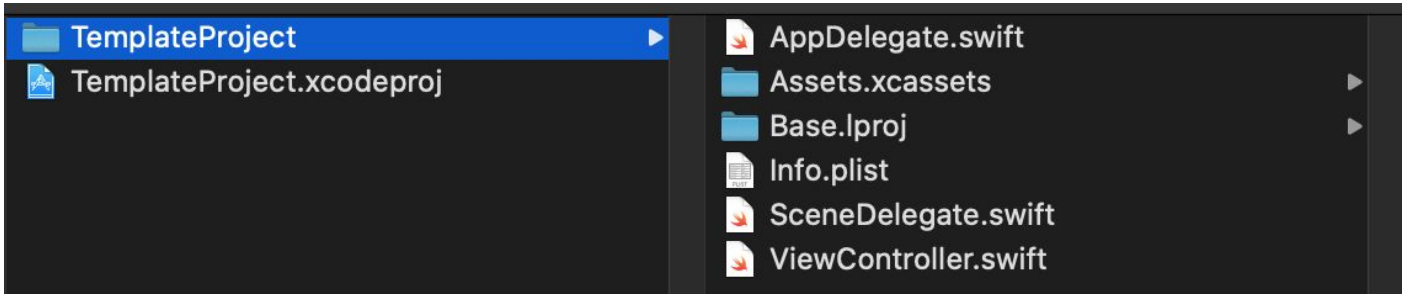


Các lựa chọn cho project mới:



New Project

- Tập tin xcodeproj là một dự án Xcode, chứa dữ liệu cấu hình dự án và liên kết đến các tệp được tham chiếu bởi dự án



- Khi mở một project, chúng ta click vào file có phần mở rộng .xcodeproj

TemplateProject.xcodeproj

- TemplateProject
  - TemplateProject
    - AppDelegate.swift
    - SceneDelegate.swift
    - ViewController.swift
    - Main.storyboard
    - Assets.xcassets
    - LaunchScreen.storyboard
    - Info.plist
  - Products

TemplateProject

General | Signing & Capabilities | Resource Tags | Info | Build Settings | Build Phases | Build Rules

PROJECT
- TemplateProject

TARGETS
- TemplateProject

Identity

Display Name: TemplateProject
Bundle Identifier: taoquynh.TemplateProject
Version: 1.0
Build: 1

Deployment Info

iOS 14.1
- [x] iPhone
- [x] iPad
- [ ] Mac

Main Interface: Main

Device Orientation
- [x] Portrait
- [ ] Upside Down
- [x] Landscape Left
- [x] Landscape Right

Status Bar Style: Default
- [ ] Hide status bar
- [ ] Requires full screen
- [ ] Supports multiple windows

App Icons and Launch Images

App Icons Source: AppIcon
Launch Screen File: LaunchScreen

Supported Intents

Class Name	Authentication

Add intents eligible for in-app handling here

Identity and Type

Name: TemplateProject
Location: Absolute
Full Path: /Users/taof/Desktop/TemplateProject/TemplateProject.xcodeproj

Project Document

Project Format: Xcode 9.3-compatible
Organization:
Class Prefix:

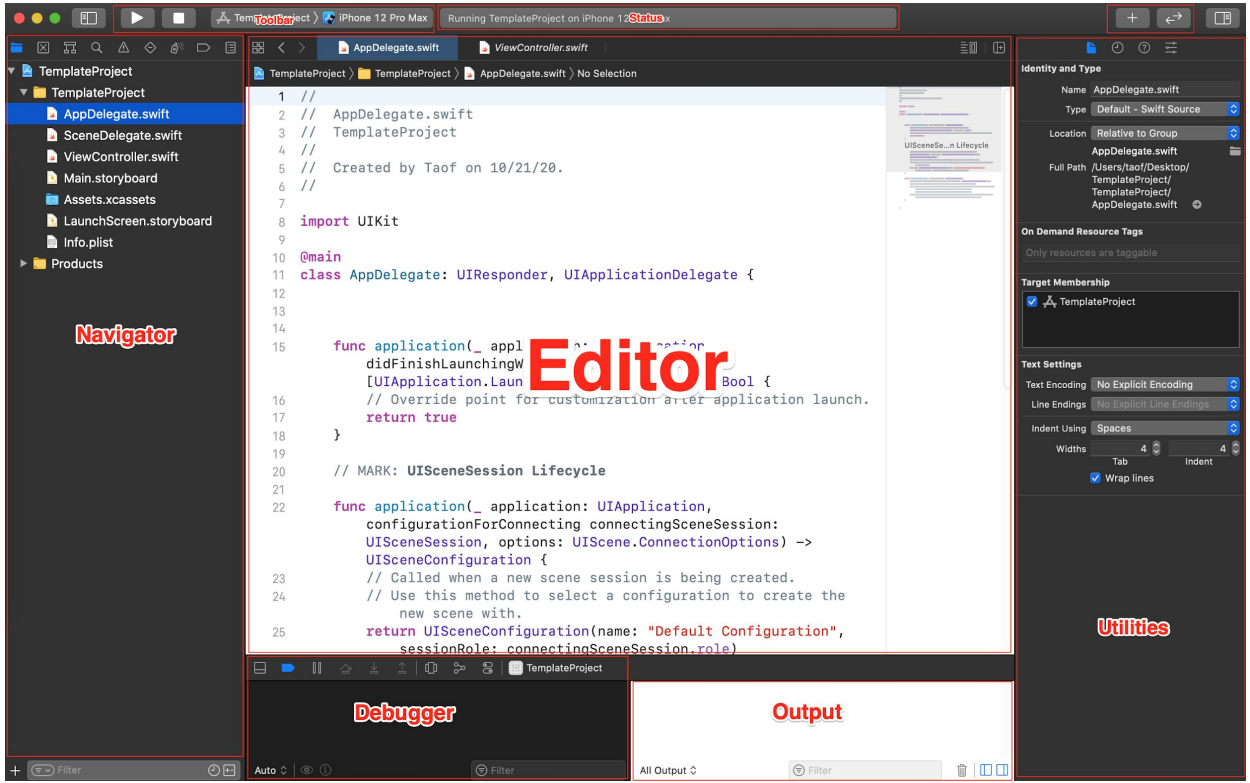
Text Settings

Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
- [x] Wrap lines

Filter

TemplateProject

Xcode Interface



AppDelegate.swift

- Class AppDelegate là điểm khởi đầu của một ứng dụng iOS
- Ứng dụng của nó **(_: didFinishLaunchingWithOptions :)** là chức năng đầu tiên mà hệ điều hành gọi khi khởi chạy ứng dụng.
- Ở AppDelegate, thường viết code ứng dụng cho iOS 12 trở về trước là:
  - Thiết lập trình điều khiển xem đầu tiên của ứng dụng (được gọi là trình điều khiển xem gốc)
  - Định cấu hình cài đặt ứng dụng và các thành phần khởi động
  - Đăng ký trình xử lý thông báo
  - Trả lời các sự kiện trong vòng đời của ứng dụng

SceneDelegate.swift

- Từ iOS 13 trở lên, SceneDelegate đảm nhận một số vai trò đại biểu của ứng dụng. Gần giống với vai trò của AppDelegate trên iOS 12 trở xuống
- Trong AppDelegate có chức năng (**_: configurationForConnecting connectingSceneSession :)** để tạo mới scene (phiên cảnh), từ đó gọi sang SceneDelegate để chạy ứng dụng
- Ở SceneDelegate, thường viết code ứng dụng cho iOS 13 trở về sau là:
    - Thiết lập trình điều khiển xem đầu tiên của ứng dụng (được gọi là trình điều khiển xem gốc)
    - Định cấu hình cài đặt ứng dụng và các thành phần khởi động
    - Đăng ký trình xử lý thông báo
    - Trả lời các sự kiện trong vòng đời của ứng dụng thông qua scene

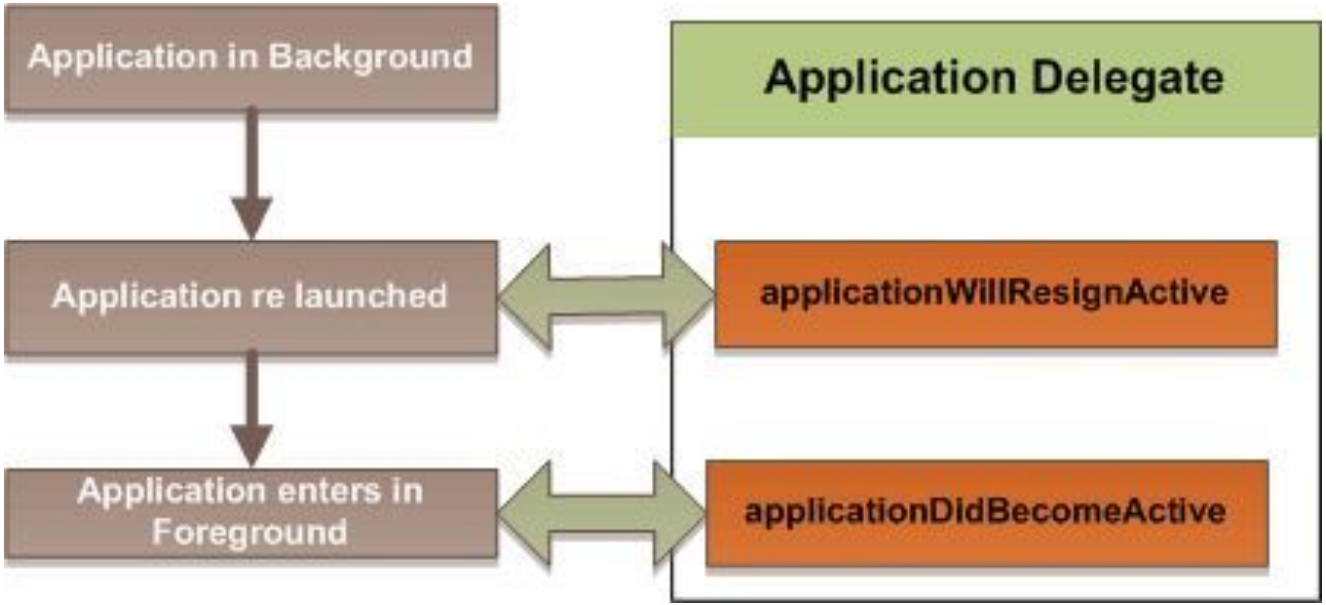
Vòng đời ứng dụng

quynh@techmaster.vn

Vòng đời của ứng dụng

- Vòng đời của 1 ứng dụng khá là quan trọng đối với các bạn developer - những người luôn muốn mang tới các trải nghiệm, phong phú, đa dạng và tốt nhất cho người dùng.

  *Khi user mở điện thoại của mình lên thì không có ứng dụng nào chạy ngoài những thứ nằm trong **OS**. **App** của bạn cũng sẽ không chạy. Sau khi user nhấn mở, **Springboard** sẽ **kích hoạt** app của bạn. App cùng với các thư viện của nó sẽ được thực thi và được tải vào bộ nhớ, trong khi đó thì Springboard sẽ nhận nhiệm vụ hiển thị màn hình **launch screen** của app. Sau cùng thì app của bạn bắt đầu được chạy và **application delegate** sẽ nhận được các **notification***


Application in Background
Application Delegate
Application re launched
applicationWillResignActive
Application enters in Foreground
applicationDidBecomeActive

Trạng thái của app

- Mọi ứng dụng iOS chạy trên các thiết bị đều có các trạng thái chuyển đổi như: **Not running, In active, Active, Background, Suspended**. Tại bất kì thời điểm nào, app của bạn đều rơi vào các trạng thái sau:

Trạng thái	Diễn tả
Not running	
Active (Foreground)	*Trạng thái khi chương trình đang hoạt động bình thường*
Inactive (Foreground)	*Tại trạng thái **Active**, nếu chương trình nhận được các sự kiện từ hệ thống như có tin nhắn mới, có điện thoại, thì ứng dụng sẽ chuyển sang trạng thái **Inactive***
Background	*Khi người dùng ấn phím **Home**, chương trình sẽ chuyển sang trạng thái **background**, ở trạng thái này ứng dụng hoàn toàn không hiển thị với người dùng, tuy nhiên các dòng lệnh vẫn được thực thi*
Suspended	*Ứng dụng nếu ở trạng thái **background** sau 5 - 10 giây thì sẽ bị "đóng băng" tạm thời, và ứng dụng chỉ thoát khỏi đóng băng khi người dùng chạy lại chương trình*

Các hàm xử lý app life-cycle trong AppDelegate

```
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.
    return true
}

func applicationWillResignActive(_ application: UIApplication) {
    // Sent when the application is about to move from active to inactive state. This can occur for certain types of temporary interruptions (such as an incoming phone call
        or SMS message) or when the user quits the application and it begins the transition to the background state.
    // Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering callbacks. Games should use this method to pause the game.
}

func applicationDidEnterBackground(_ application: UIApplication) {
    // Use this method to release shared resources, save user data, invalidate timers, and store enough application state information to restore your application to its
        current state in case it is terminated later.
    // If your application supports background execution, this method is called instead of applicationWillTerminate: when the user quits.
}

func applicationWillEnterForeground(_ application: UIApplication) {
    // Called as part of the transition from the background to the active state; here you can undo many of the changes made on entering the background.
}

func applicationDidBecomeActive(_ application: UIApplication) {
    // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application was previously in the background, optionally refresh
        the user interface.
}

func applicationWillTerminate(_ application: UIApplication) {
    // Called when the application is about to terminate. Save data if appropriate. See also applicationDidEnterBackground:.
}
```

Các hàm xử lý app life-cycle trong SceneDelegate

```swift
func scene(_ scene: UIScene, willConnectTo session: UISceneSession, options connectionOptions: UIScene.ConnectionOptions) {
    // Use this method to optionally configure and attach the UIWindow `window` to the provided UIWindowScene `scene`.
    // If using a storyboard, the `window` property will automatically be initialized and attached to the scene.
    // This delegate does not imply the connecting scene or session are new (see `application:configurationForConnectingSceneSession` instead).
    guard let _ = (scene as? UIWindowScene) else { return }
}

func sceneDidDisconnect(_ scene: UIScene) {
    // Called as the scene is being released by the system.
    // This occurs shortly after the scene enters the background, or when its session is discarded.
    // Release any resources associated with this scene that can be re-created the next time the scene connects.
    // The scene may re-connect later, as its session was not necessarily discarded (see `application:didDiscardSceneSessions` instead).
}

func sceneDidBecomeActive(_ scene: UIScene) {
    // Called when the scene has moved from an inactive state to an active state.
    // Use this method to restart any tasks that were paused (or not yet started) when the scene was inactive.
}

func sceneWillResignActive(_ scene: UIScene) {
    // Called when the scene will move from an active state to an inactive state.
    // This may occur due to temporary interruptions (ex. an incoming phone call).
}

func sceneWillEnterForeground(_ scene: UIScene) {
    // Called as the scene transitions from the background to the foreground.
    // Use this method to undo the changes made on entering the background.
}

func sceneDidEnterBackground(_ scene: UIScene) {
    // Called as the scene transitions from the foreground to the background.
    // Use this method to save data, release shared resources, and store enough scene-specific state information
    // to restore the scene back to its current state.
}
```

Vòng đời UIViewController

quynh@techmaster.vn

UIViewControler

Một thể hiện của class UIViewController mình tạm gọi là ViewController

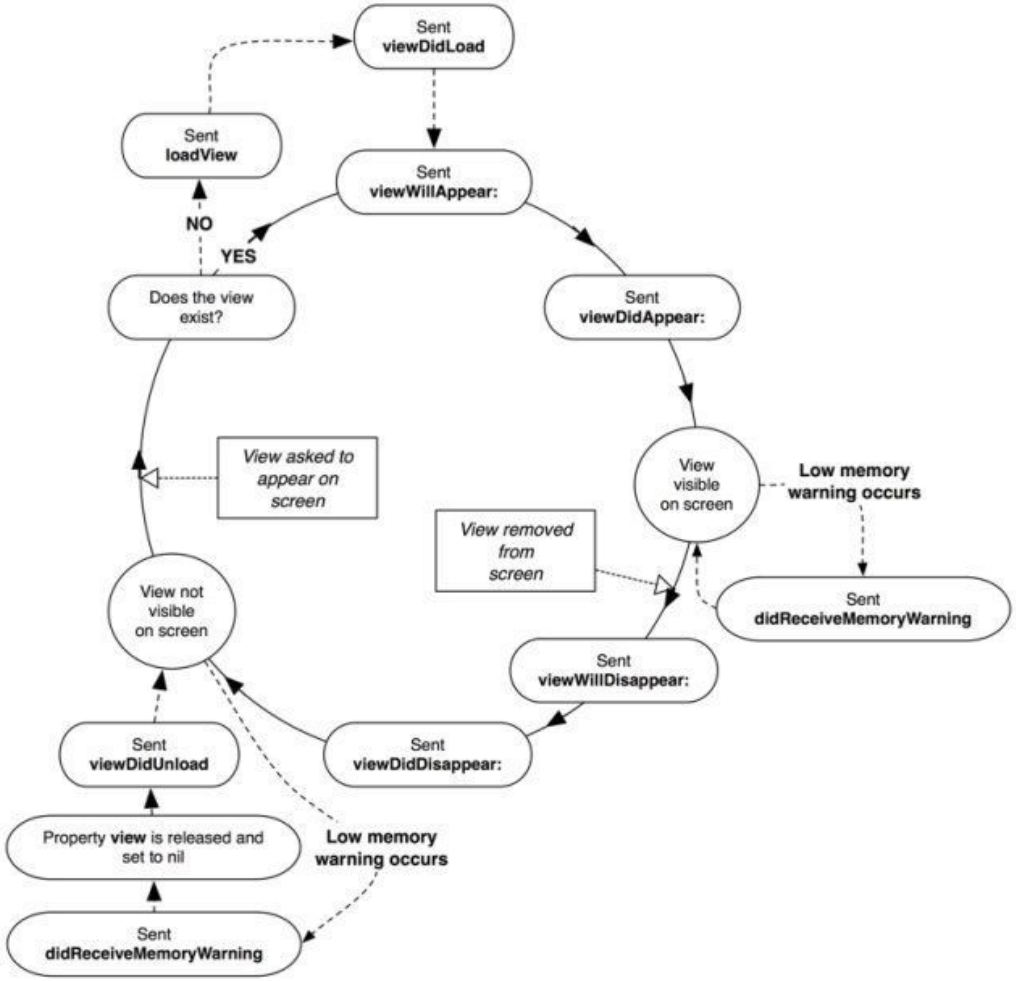
- Một **ViewController** được hiểu gần như là một màn của ứng dụng
- Một trong những vai trò quan trọng của **ViewController** là quản lý các **View** hiển thị nội dung trong App
- **ViewController** có thể giao tiếp và kết hợp với các **ViewController** khác
- **ViewController** cung cấp nhiều hành vi (**behaviour**) phổ biến cho tất cả các ứng dụng iOS. Thông thường, những hành vi này được xây dựng vào các lớp cơ sở (**base class**).

Vòng đời của UIViewController

- Vòng đời của một View Controller là được tính từ lúc nó được nạp vào bộ nhớ (**RAM**) cho tới khi nó bị huỷ khỏi bộ nhớ.

- Để huỷ **ViewController**, chúng ta cần gọi hàm **deinit()**

- Sau đây là sơ đồ nói về hoạt động sống của một **ViewController** cùng với các hàm nó sẽ được gọi tại một thời điểm xác định nào đó


Sent
viewDidLoad
Sent
loadView
Sent
viewWillAppear:
NO
YES
Does the view
exist?
Sent
viewDidAppear:
View asked to
appear on
screen
View
visible
on screen
Low memory
warning occurs
View removed
from
screen
View not
visible
on screen
Sent
didReceiveMemoryWarning
Sent
viewWillDisappear:
Sent
viewDidUnload
Sent
viewDidDisappear:
Property view is released and
set to nil
Low memory
warning occurs
Sent
didReceiveMemoryWarning

Trạng thái hoạt động của ViewController

loadView

- Hàm này sẽ được gọi đầu tiên khi khởi tạo một *View Controller*.

viewDidLoad

- Khi *View Controller* đã được nạp vào bộ nhớ, thì hàm **viewDidLoad** được gọi.
- Chỉ được gọi một lần duy nhất trong chu kỳ sống của view đó thôi
- Thường dùng để chuẩn bị data hoặc là khởi tạo các giá trị mặc định cho các object cũng như UI trên màn hình.

viewWillAppear

- Hàm sẽ được gọi trước khi một view được thêm vào hệ thống view và trước animation hiển thị một view.

viewDidAppear

- Hàm sẽ được gọi khi một view đã được thêm vài hệ thống view và đã hiển thị lên màn hình.

viewWillDisappear

- Gọi khi một view đã được ẩn khỏi màn hình và amination khi ẩn view đó.

viewDidDisappear

- Gọi khi một view đã được ẩn khỏi màn hình.

Lưu ý khi load dữ liệu tại các hàm

- Nếu dữ liệu là dạng tĩnh và **không thay đổi nhiều**, chúng ta nên load chúng ở phương thức **viewDidlLoad**.

- Nếu dữ liệu là động và thay đổi thường xuyên, chúng ta nên load tại phương thức **viewDidAppear**.

- Lưu ý là tại 2 phương thức, dữ liệu cần phải load theo dạng bất đồng bộ ở một **thread** khác để tránh ảnh hưởng đến **UI**.

```
10
11 class ViewController: UIViewController {
12
13     override func viewDidLoad() {
14         super.viewDidLoad()
15         // Do any additional setup after loading the view.
16         print("viewDidLoad")
17     }
18
19     override func loadView() {
20         super.loadView()
21         print("load View")
22     }
23
24     override func viewWillAppear(_ animated: Bool) {
25         super.viewWillAppear(true)
26         print("view Will Appear")
27     }
28
29     override func viewDidAppear(_ animated: Bool) {
30         super.viewDidAppear(true)
31         print("view Did Appear")
32     }
33
34     override func viewWillDisappear(_ animated: Bool) {
35         super.viewWillDisappear(true)
36         print("view Will Disappear")
37     }
38
39     override func viewDidDisappear(_ animated: Bool) {
40         super.viewDidDisappear(true)
41         print("view Did Disappear")
42     }
43 }
```

MyProjectController

```
load View
viewDidLoad
view Will Appear
view Did Appear
```

Auto | Filter | All Output | Filter