BOOSTCAMP iOS

남상욱

1. 자신이 터득한 오토레이아웃 구성 노하우

2. IBOutlet과 IBAction을 사용할 때 주의할 점들

Understanding Auto Layout

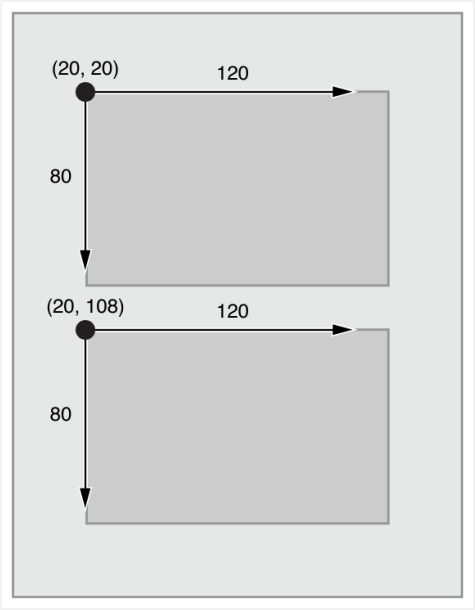
Auto Layout dynamically calculates the size and position of all the views in your view hierarchy, based on constraints placed on those views. For example, you can constrain a button so that it is horizontally centered with an Image view and so that the button's top edge always remains 8 points below the image's bottom. If the image view's size or position changes, the button's position automatically adjusts to match.

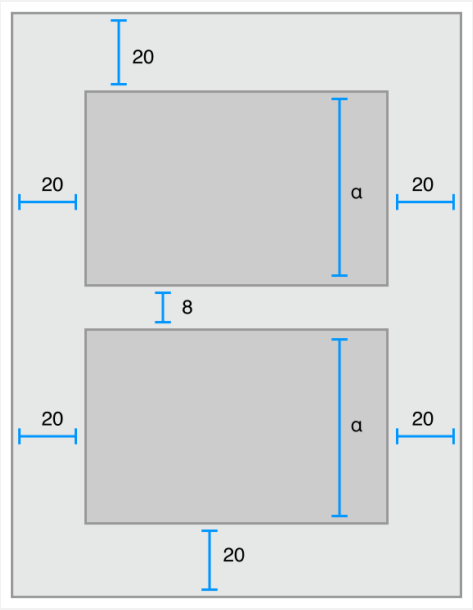
This constraint-based approach to design allows you to build user interfaces that dynamically respond to both internal and external changes.

→ 자동으로 View의 크기와 위치를 계산해주는 것.

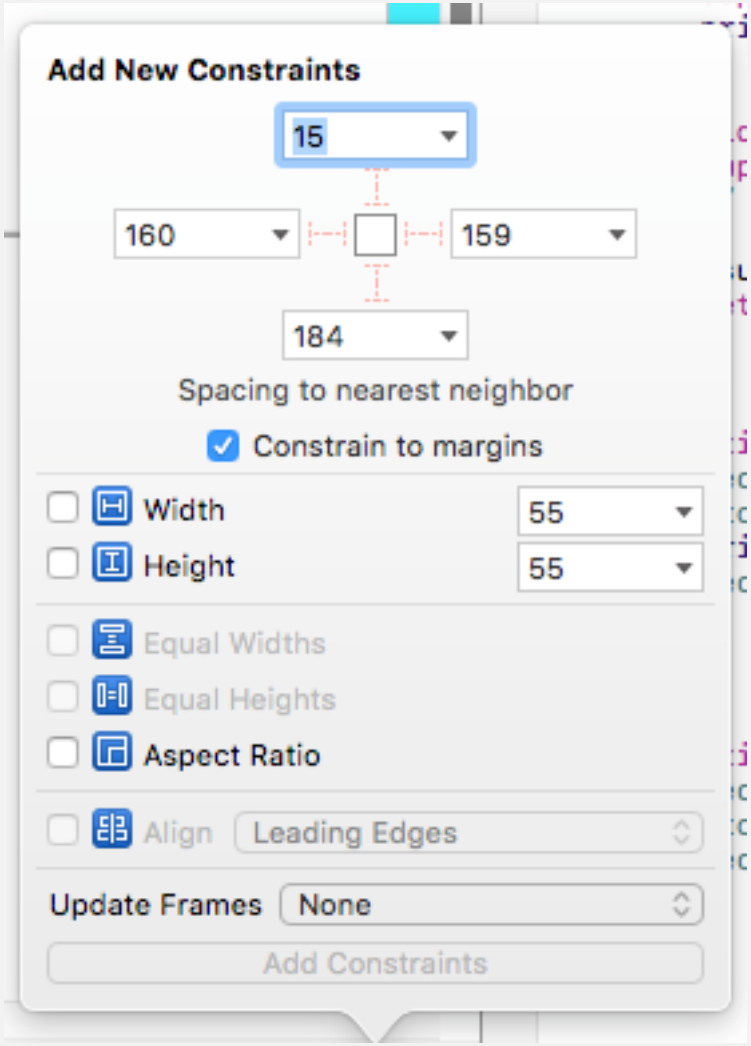
→ iPhone 4S, iPhone 5S, iPhone 6 Plus, iPad와 같은 다양한 화면과 디바이스 스크린의 회전

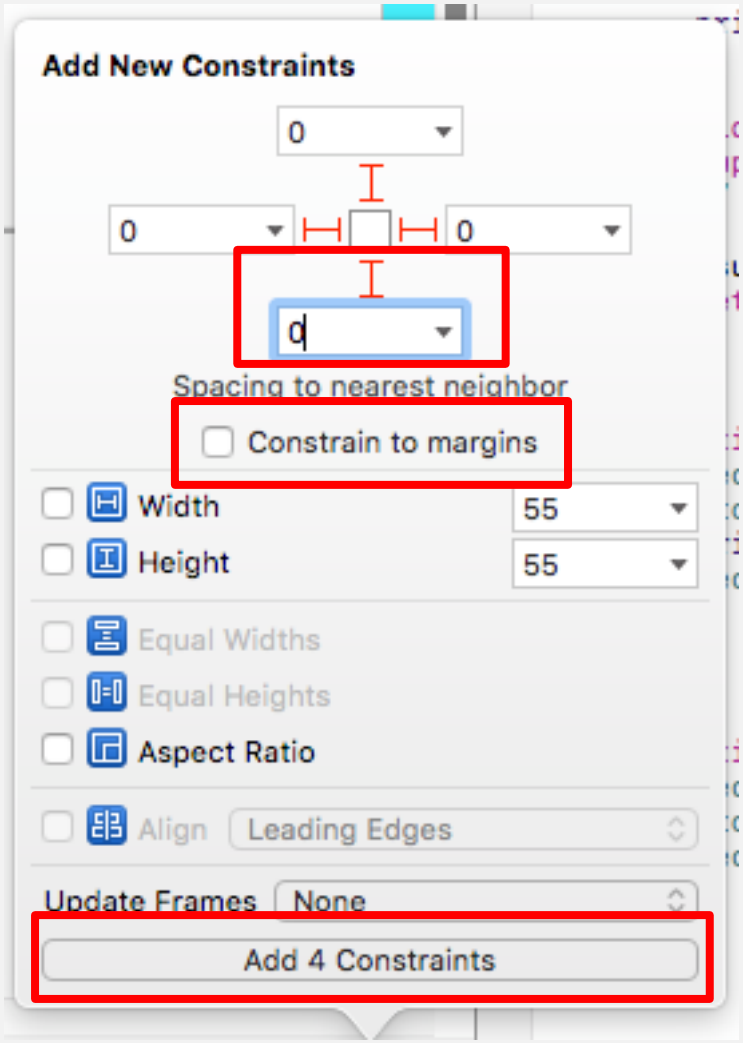
→ 이미지 크기, 나라 언어의 차이, 글꼴 크기

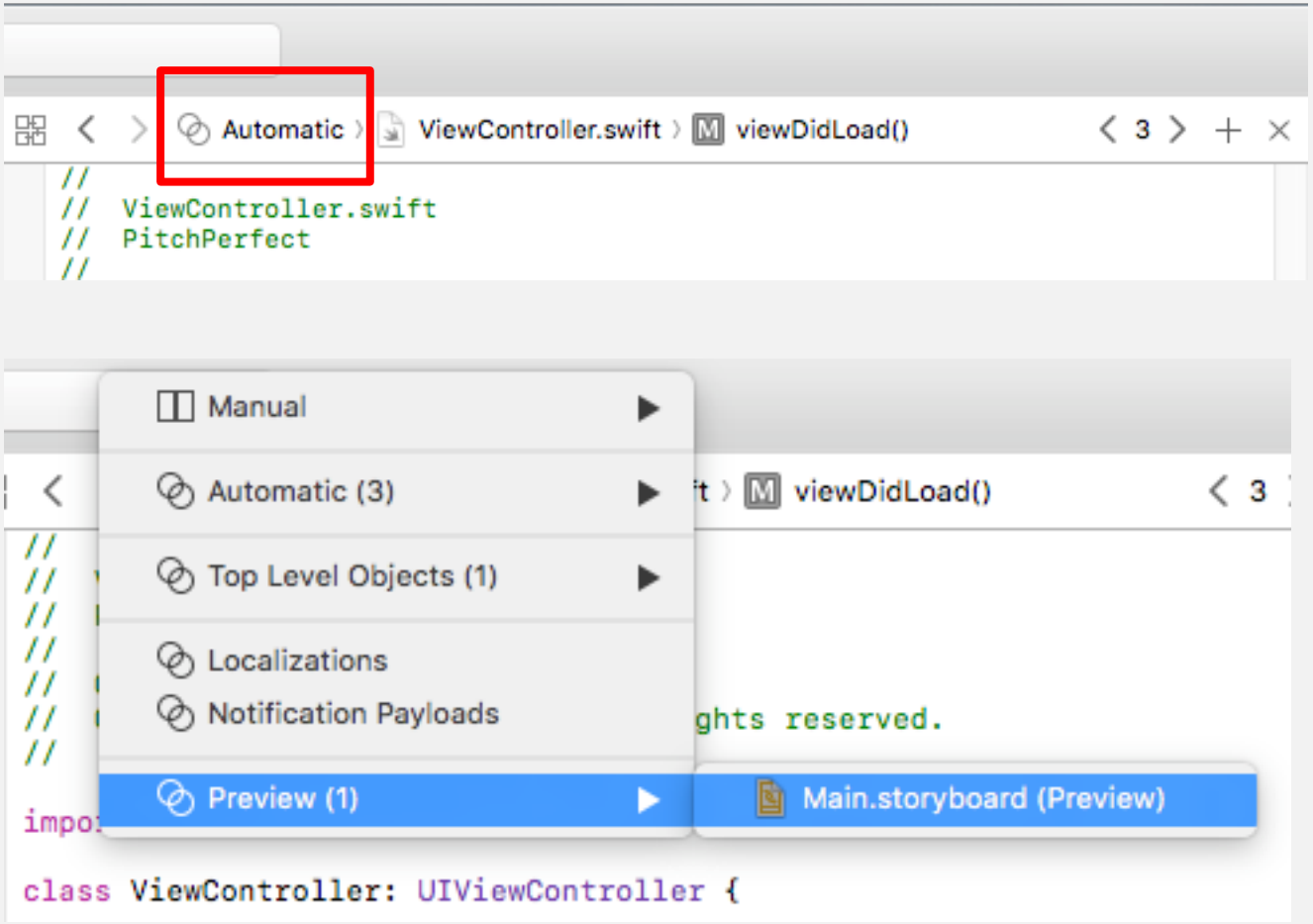




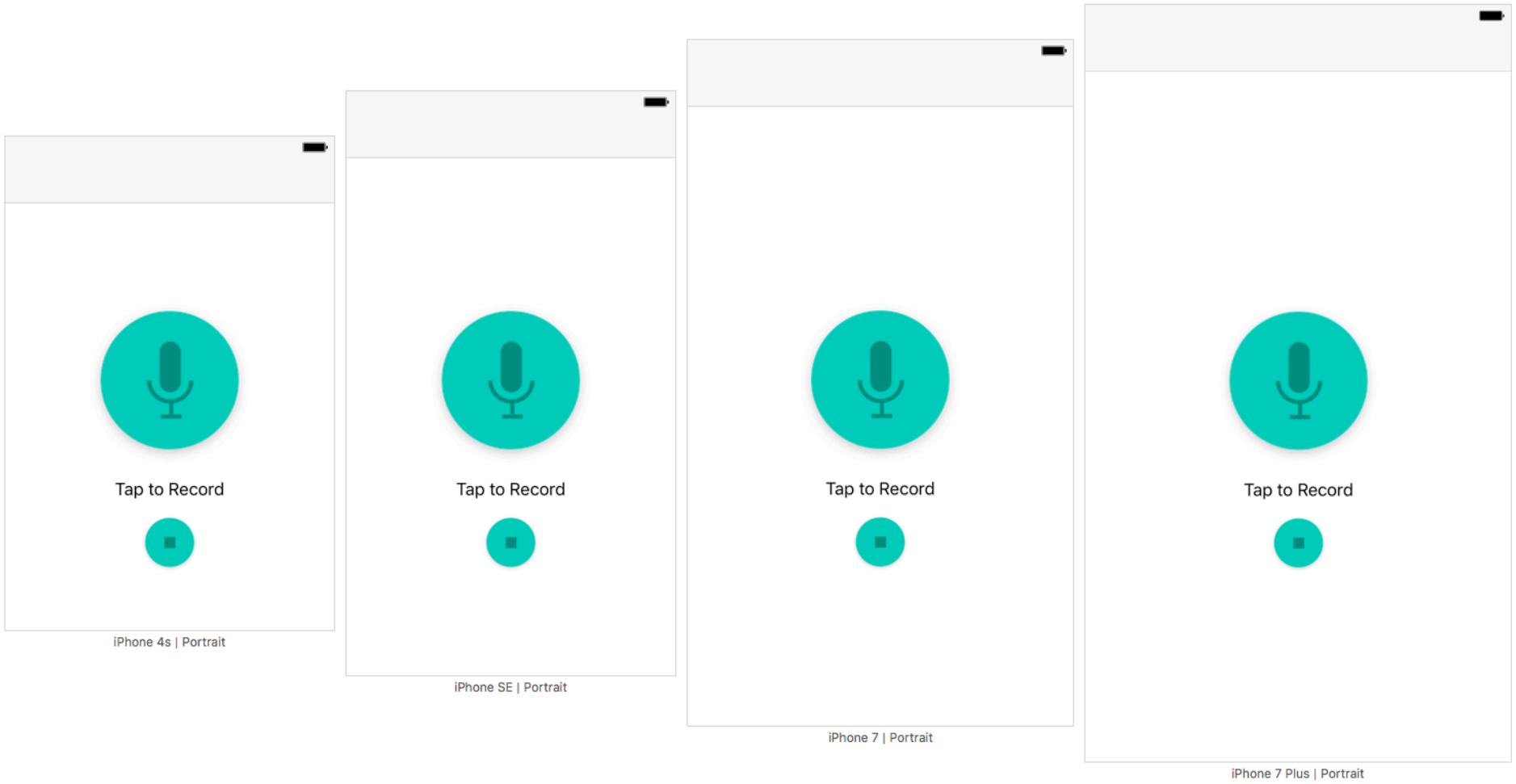
→ Auto Layout defines your user interface using a series of constraints. Constraints typically represent a relationship between two views. Auto Layout then calculates the size and location of each view based on these constraints. This produces layouts that dynamically respond to both internal and external changes.


Add New Constraints
15
160
159
184
Spacing to nearest neighbor
Constrain to margins
Width
55
Height
55
Equal Widths
Equal Heights
Aspect Ratio
Align
Leading Edges
Update Frames
None
Add Constraints


Add New Constraints
0
0
0
0
Spacing to nearest neighbor
Constrain to margins
Width
55
Height
55
Equal Widths
Equal Heights
Aspect Ratio
Align
Leading Edges
Update Frames
None
Add 4 Constraints



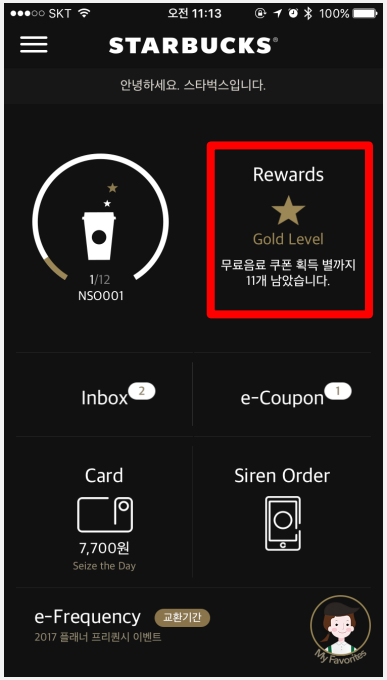
→ Show Assistant Editor - Automatic - Preview

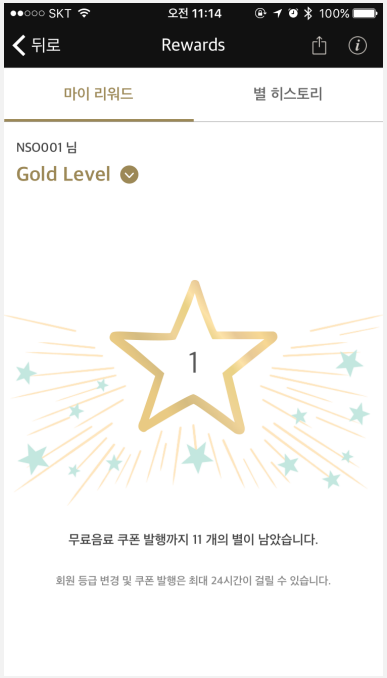


→ Show Assistant Editor - Automatic - Preview

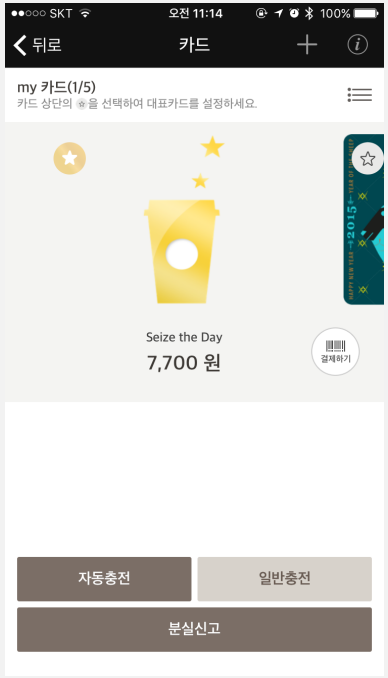
3. 자신이 자주 사용하는 앱의 여러 ViewController에서 라이프사이클에 따라 어떤 동작들이 이루어지고 있을지 상상해보기

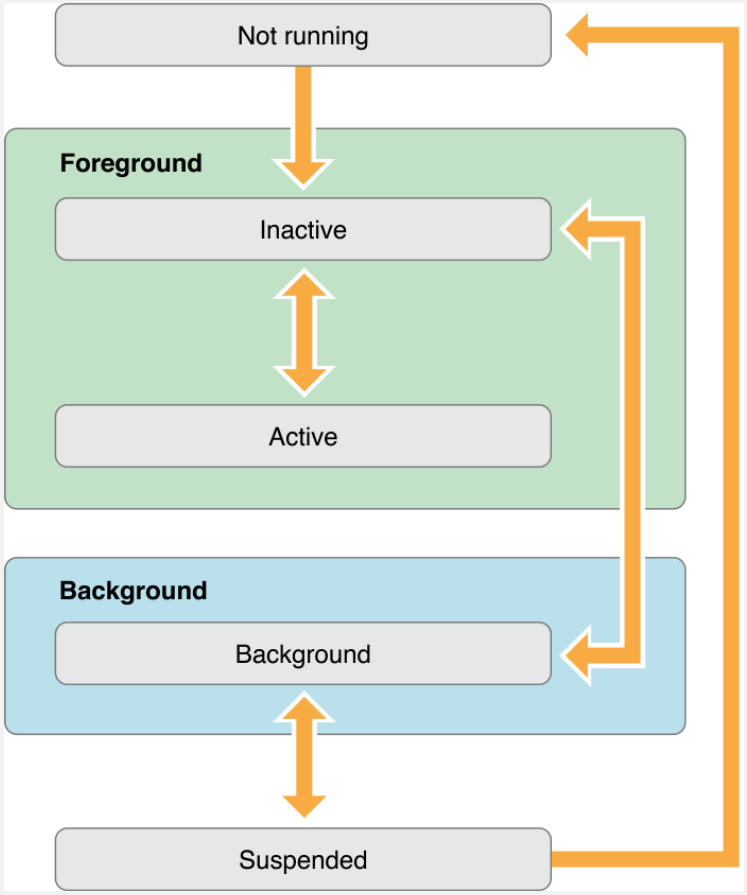
4. 자신이 사용하는 앱의 여러 ViewController에서 MVC 디자인 패턴을 따라 어떻게 클래스가 구성되어 있을지 상상해보기


SKT 오전 11:13 100%
STARBUCKS
안녕하세요. 스타벅스입니다.
1/12
NSO001
Rewards
Gold Level
무료음료 쿠폰 획득 별까지
11개 남았습니다.
Inbox 2
e-Coupon 1
Card
7,700원
Seize the Day
Siren Order
e-Frequency 교환기간
2017 플래너 프리퀀시 이벤트


SKT 오전 11:14 100%
뒤로
Rewards
마이 리워드
별 히스토리
NSO001 님
Gold Level
1
무료음료 쿠폰 발행까지 11 개의 별이 남았습니다.
회원 등급 변경 및 쿠폰 발행은 최대 24시간이 걸릴 수 있습니다.


SKT 오전 11:14 100%
뒤로
Inbox
스타벅스의 새로운 소식과 주요 알림 사항을 볼 수 있는 서비스입니다.
전체선택
선택삭제
소식
활동
54 기간 내, 2017년 신년 카드 2종을 계정에 등록 시, 2,017명 추첨을 통해 리액션 BOGO쿠폰 한 장을 드립니다. 2017-01-06 00:00:00
53 e-프리퀀시 이벤트는 2017년 1월 7일까지이며, 기한 내 가까운 매장을 방문하시어 교환 받으시기 바랍니다 (1월 7일 이후 스티커 소멸예정) 2017-01-04 14:47:00
52 2017년에는 전자 영수증으로 바꾸고 다양한 행운도 받아보세요! 2017-01-04 00:00:00
51 2017 뉴이어, 스타벅스와 함께 즐겁고 활기차게 시작하세요! 2017-01-01 11:00:00
50 Tri Star Dash! 스타벅스 신규음료 드시면, 별 3개 추가로 적립해 드립니다. 2017-01-01 11:00:00


SKT 오전 11:14 100%
뒤로
카드
my 카드(1/5)
카드 상단의 ☆을 선택하여 대표카드를 설정하세요.
Seize the Day
7,700 원
결제하기
자동충전
일반충전
분실신고

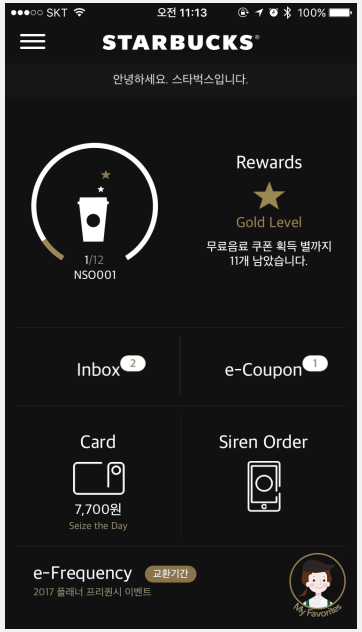


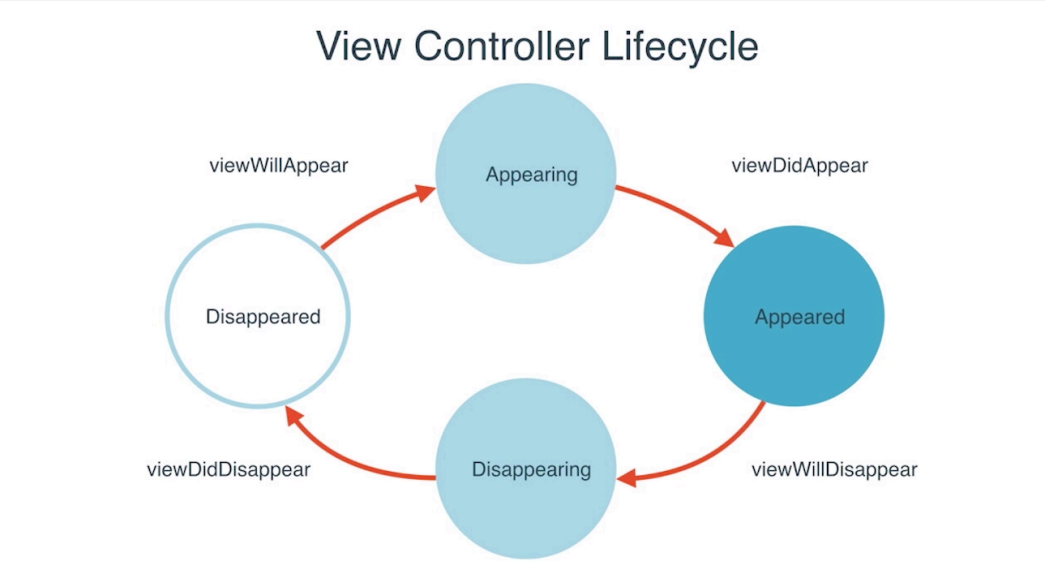
- Not Running: 앱이 실행되지 않은 상태
  (Inactive와 Active → Foreground)
- Inactive: 앱이 실행중인 상태 그러나 아무런 이벤트를 받지 않는 상태
- Active: 앱이 실행중이며 이벤트가 발생한 상태
- Background: 앱이 백그라운드에 있는 상태 그러나 실행되는 코드가 있는 상태
- Suspened: 앱이 백그라운드에 있고 실행되는 코드가 없는 상태

```swift
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
        [UIApplicationLaunchOptionsKey: 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:.
    }


}
```

<AppDelegate.swift>

- application(_:didFinishLaunching:)
  - 앱이 처음 시작될 때 실행
- applicationWillResignActive
  - Active 에서 Inactive로 이동될 때 실행
- applicationDidEnterBackground
  - background 상태일 때 실행
- applicationWillEnterForeground
  - background에서 foreground로 이동 시
  (아직 foreground에서 실행중이진 않음)
- applicationDidBecomeActive
  - Active상태가 되어 실행 중일 때
- applicationWillTerminate
  - 앱이 종료될 때 실행





- **ViewDidLoad** - 해당 뷰컨트롤러 클래스가 생성될 때 딱 한번만 실행되기 때문에 초기화 할 때 사용
  - UIView를 상속받는 UIButton, UIImageView, UITableView 등
- **ViewWillAppear** - 뷰 컨트롤러가 화면에 나타나기 직전에 실행됩니다.
- **ViewDidAppear** - 뷰 컨트롤러가 화면에 나타난 직후에 실행됩니다.
  - 화면에 적용될 애니메이션을 그려줍니다.
  - DB서버의 API 형식으로 제공받는 정보를 받아와 화면을 업데이트
- **ViewWillDisappear** - 해당 뷰 컨트롤러가 사라지기 직전에 실행됩니다.
- **viewDidDisappear** - 해당 뷰 컨트롤러가 사라졌을 때 실행됩니다.
  - 해당 화면에서 생성된 자원(마이크, DB커넥션 등)을 해제하는 로직이 실행됩니다.

5. 프로그래밍에서 디자인 패턴이란 무엇인가, 디자인 패턴이 가지는 의미는 무엇일까

6. 네비게이션 컨트롤러의 동작은 자료구조의 형태에서 어떤 자료구조 방법론과 유사할까? iOS에서 네비게이션 컨트롤러를 사용하지 않고 화면을 전환하는 다른 방법에는 어떤 것들이 있을까?

디자인 패턴이란?

프로그램 개발에서 자주 나타나는 과제를 해결하기 위한 방법 중 하나로, 소프트웨어 개발과정에서 발견된 Know-How를 축적하여 이름을 붙여 이후에 재사용하기 좋은 형태로 특정 규약을 묶어서 정리한 것.

→ iOS Framework는 MVC와 Delegation이라는 디자인 패턴에 의존

수시로 변경되는 요구사항

→ 요구사항 변경에 대한 Source Code 수정 최소화

여러 사람이 같이 하는 팀 프로젝트 진행

→ 범용적인 코딩 스타일 적용이 가능하다.

→ 좋은 가독성(직관적인 코드)을 볼 수 있다.

Thank u