

Storyboard

- Xcode에서 UI와 UX를 손쉽게 구성할 수 있도록 마련한 도구
- 화면의 구성과 흐름을 파악할 수 있다
- 코드와 직접 아울렛 및 액션을 연결해 줄 수 있다

IBOutlet / IBAction

- IB(interface bulider)와 코드를 연결해 줄수있는 IB를 위한 키워드
- main storyboard와 연결

Interface Builder

- 과거 xcode에는 IB가 내장되어있지 않았다.

Object Library

- IB에 올릴수 있는 객체의 모음
- IB 우측 하단
- Firter창에 검색 가능

IBOutlet

- 클래스의 UI관련 **프로퍼티**와 **IB(스토리보드 또는 XIB)의 객체**와 연결할수있도록 도와주는 키워드

```
//스토리보드의 뷰들과 컨트롤러를 연결
@property IBOutlet UILabel *mylabel; //라벨
@property IBOutlet UIButton *mybutton; //버튼
@property IBOutlet UISegmentedControl *mycontrol; //세그먼트
@property IBOutlet UITextField *mytext; //텍스트필드
@property IBOutlet UISlider *myslider; //슬라이드
@property IBOutlet UISwitch *myswitch; //스위치
@property IBOutlet UIStepper *mystepper; //스태퍼

@property (weak, nonatomic) IBOutlet UIView *changeview; //색상이 바뀌는 뷰
@property (weak, nonatomic) IBOutlet UISlider *redslider; //빨강 슬라이드
@property (weak, nonatomic) IBOutlet UISlider *greenslider; //초록 슬라이드
@property (weak, nonatomic) IBOutlet UISlider *blueslider; //파랑 슬라이드
```

IBAction

- 클래스의 특정 메서드와 스토리보드 객체의 이벤트(액션)와 연결해 줄수있도록 도와주는 키워드

```
//스토리보드의 버튼의 액션연결
- (IBAction)clickbutton:(id)sender
{
    //sender가 버튼일 경우
    //isKindOfClass는 해당 클래스가 맞느냐
    if ([sender isKindOfClass:[UIButton class]]) {
        UIButton *button = sender;
        NSLog(@"button title : %@, tag : %ld", [[button titleLabel]text], [button tag]);
        //해당하는 버튼이 아닐 경우
    }else {
        NSLog(@"버튼이아님");
    }
}
```

MVC - Model, View, Controller

- Model, View, Controller 모델
- 프로그래밍 디자인 패턴의 하나
- 자료저장소, 뷰, 컨트롤러를 분리하여 구현하는 형태

정리사항

- 프로그램을 만들고, 문제를 해결하는데 유용한 패턴 = 디자인패턴, mvc
- 프로퍼티 **IBOutlet**, 1 : 1
- 메서드 **IBAction**, 1 : N, 반환타입이 void와 동일시
- NIP / 인터페이스 빌더
- is initial view controller / 뷰 진입점 표시
- main interface / 앱 실행후 처음 나타나는 화면
- 스토리보드의 뷰가 어떤 컨트롤러를 상속받는지 정해주어야 한다.
- View / 디스플레이만 주관, 무조건 보여지는것만, 화면에 직접적으로 표시, UIView
- Controller / 기능적인 부분 관리, 기능을 하기위해서는 무조건 view가 필요하다!, 사용자와 데이터사이의 중재자, 화면에 보이지않음, UINavigationController
- Model / 데이터를 관리, DB, DataCenter