

# 4Day\_Summary

Date @Jan 13, 2020

## Delegate

→ **위임하다** 라는 의미 그대로, Protocol(=Interface)처리 해야 **할 일 중 일부를 다른 객체에게 넘기는 것**을 뜻한다.

쉽게 생각해서 타노스가 타임스톤의 기능을 건틀릿에 장착했을때 시간을 조작하는 능력을 얻듯이,  
원하는 델리게이트를 장착하면 해당 기능을 사용할 수 있게된다ㅋㅋ

지난번에 UITextField를 배웠기 때문에 UITextField의 Delegate를 어떻게 사용해야되는지 알아보도록 하겠습니다.

### <사용법>

1. delegate 채택 작업(예: UITextFieldDelegate, UITableViewDelegate)

```
// 1: 채택
extension ViewController: UITextFieldDelegate {

}
```

2. Delegate의 기능을 어떤 ViewController가 쓸건지 정해준다.

```
override func viewDidLoad() {
    super.viewDidLoad()

    // 2: delegate 사용
    textField.delegate = self
    //textField의 델리게이트 기능을 self(ViewController)가 쓰겠다.
}
```

3. Delegate기능 사용하기 가져와서 사용하기

→ UITextFieldDelegate의 문서를 들어가보면 많은 기능들이 있다.

→ 필요한 기능이 있으면 찾아서 쓰면된다!

```
public protocol UITextFieldDelegate : NSObjectProtocol {

    @available(iOS 2.0, *)
    optional func textFieldShouldBeginEditing(_ textField: UITextField) -> Bool // return NO to disallow editing.

    @available(iOS 2.0, *)
    optional func textFieldDidBeginEditing(_ textField: UITextField) // became first responder

    @available(iOS 2.0, *)
    optional func textFieldShouldEndEditing(_ textField: UITextField) -> Bool // return YES to allow editing to stop and to res

    @available(iOS 2.0, *)
    optional func textFieldDidEndEditing(_ textField: UITextField) // may be called if forced even if shouldEndEditing returns

    @available(iOS 10.0, *)
    optional func textFieldDidEndEditing(_ textField: UITextField, reason: UITextField.DidEndEditingReason) // if implemented,

    @available(iOS 2.0, *)
    optional func textField(_ textField: UITextField, shouldChangeCharactersIn range: NSRange, replacementString string: String) -> Bool?

    @available(iOS 13.0, *)
    optional func textFieldDidChangeSelection(_ textField: UITextField)
```

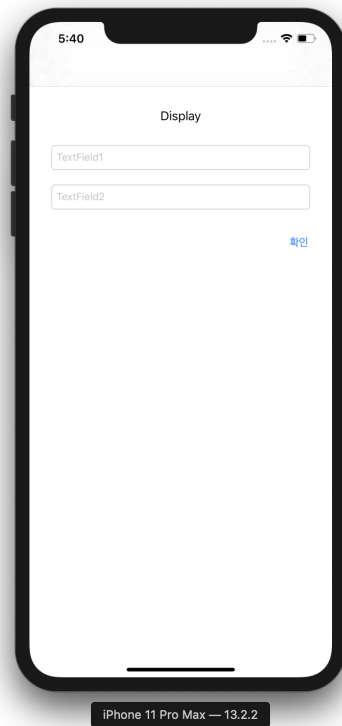
```

@available(iOS 2.0, *)
optional func textFieldShouldClear(_ textField: UITextField) -> Bool // called when clear button pressed. return NO to ignore

@available(iOS 2.0, *)
optional func textFieldShouldReturn(_ textField: UITextField) -> Bool // called when 'return' key pressed. return NO to ignore
}

```

## <만들 화면>



```

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var displayLabel: UILabel!
    @IBOutlet weak var textField1: UITextField!
    @IBOutlet weak var textField2: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()

        // 2 delegate 채택
        textField1.delegate = self
        textField2.delegate = self
    }

}

// 1: 채택
extension ViewController: UITextFieldDelegate {

}

```

- 사용자가 텍스트 필드에 텍스트를 입력하고 **Return**을 누를 때마다 **displayLabel**에 표시해주기

→ **textFieldShouldReturn**메서드는 textField에서 Return을 누를때마다 호출되는 메서드입니다.

```
extension ViewController: UITextFieldDelegate {  
  
    // 3: 델리게이트 기능 사용하기  
  
    func textFieldShouldReturn(_ textField: UITextField) -> Bool {  
        displayLabel.text = textField1.text  
  
        return true  
    }  
}
```

- 사용자가 텍스트 필드에 텍스트를 입력할때 실시간으로 displayLabel에 표시되는 방법

→ **shouldChangeCharactersIn**메서드는 textField의 text를 입력할때마다 실시간으로 호출되는 메서드

→ 실시간으로 텍스트를 호출하고 싶을때 많이 사용

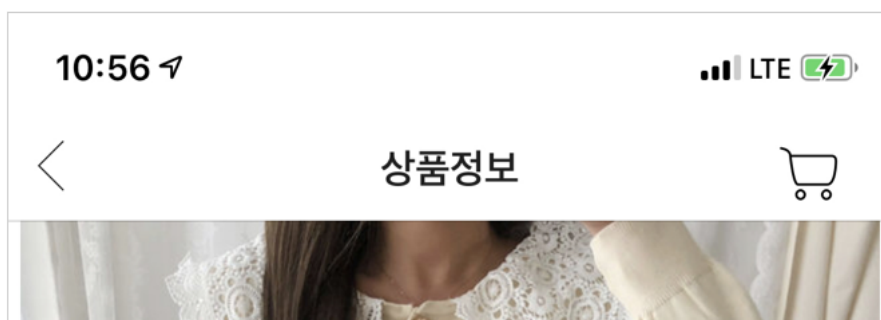
```
func textField(_ textField: UITextField, shouldChangeCharactersIn range: NSRange, replacementString string: String) -> Bool {  
  
    // 어떤식으로 파라미터 데이터가 넘어오는지 확인해보기  
    print(string)  
    print(textField)  
    print(range)  
  
    // 방법 1  
    let text = (textField.text ?? "") + string  
    displayLabel.text = text  
  
    // 방법 2  
    //if let text = textField.text {  
    //    self.displayLabel.text = text + string  
    //} else {  
    //  
    //}  
  
    return true  
}
```

UITextField의 Delegate에는 많은 기능들이 있는데, 메서드를 보면 쉽게 명명되어있다.

하나씩 살펴보기

## Navigation Controller

→ 화면을 스택(Stack)으로 관리하고, 아래의 화면처럼 버튼, 타이틀 형식의 기능이 있는 Controller

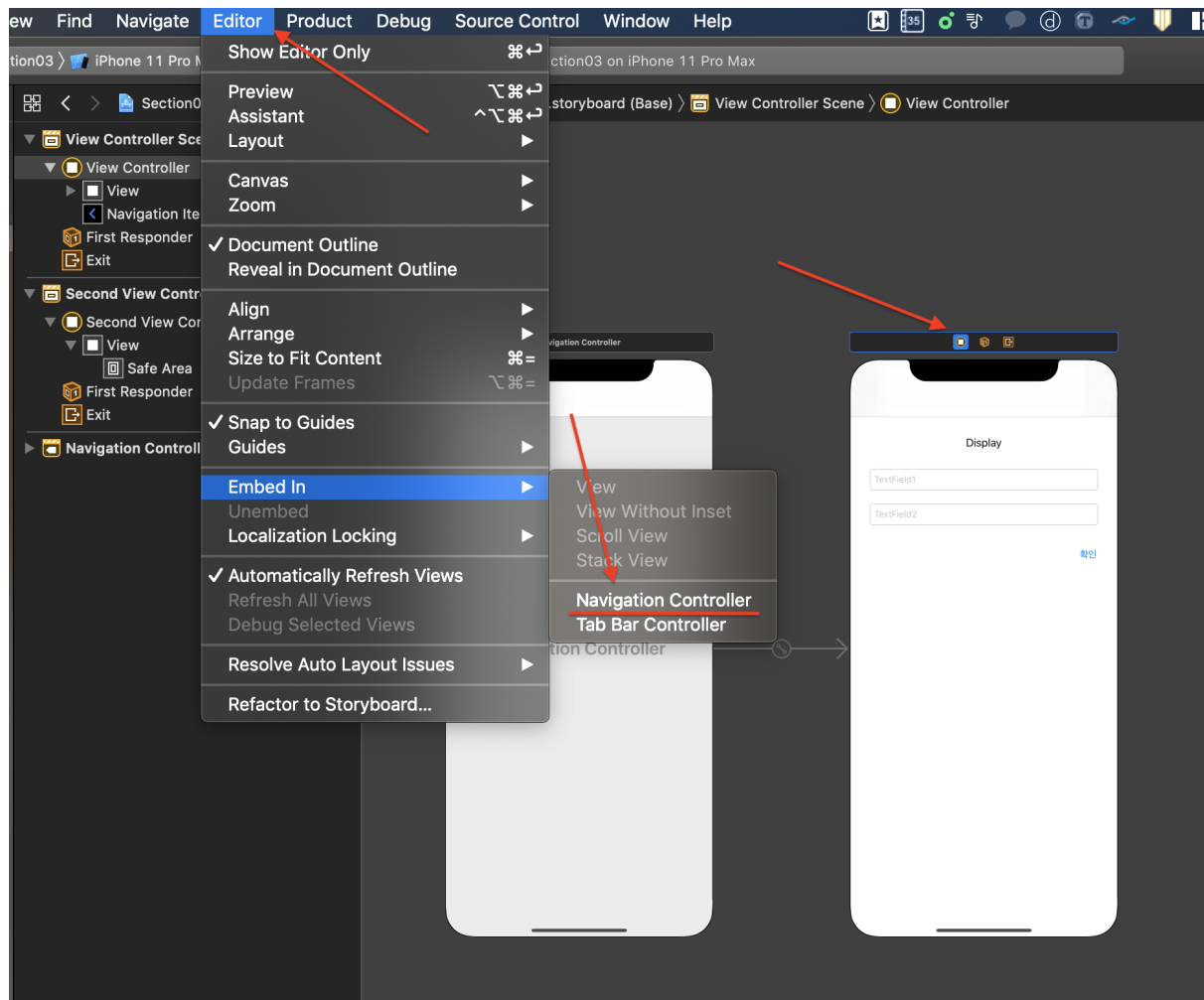


화면 이동 방법에 대해 크게 **2가지**가 있다고 했습니다.

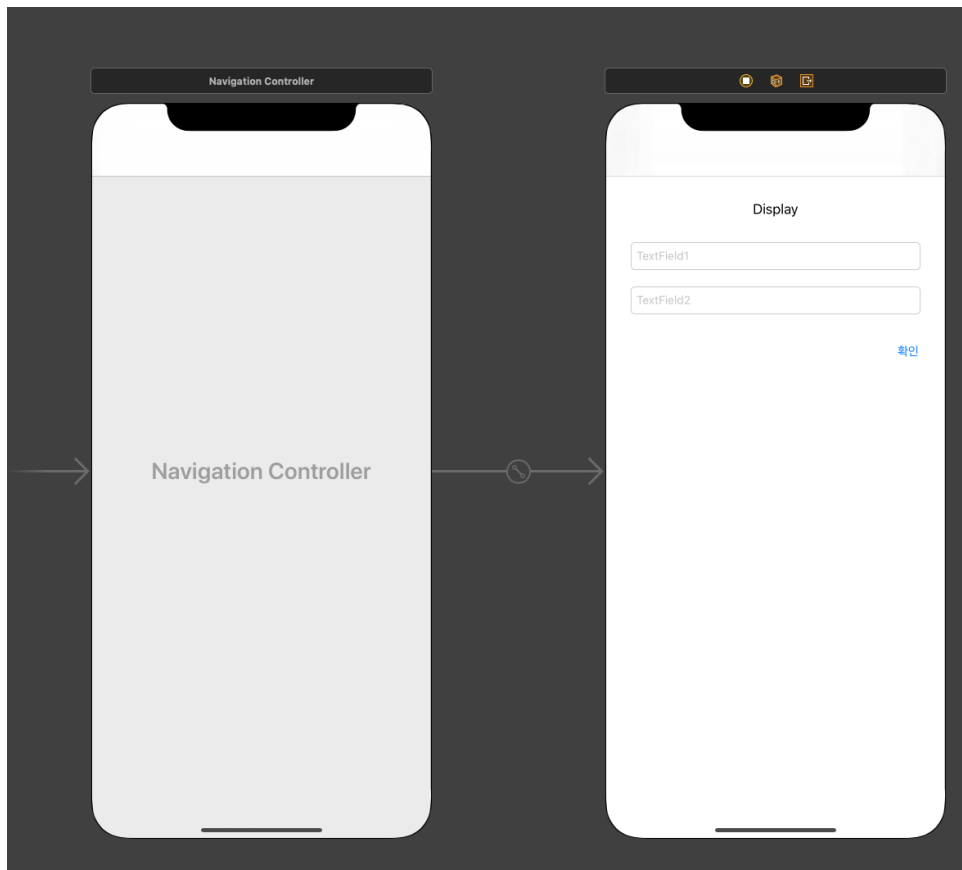
처음에 배웠던 모달형식의 **present ↔ dismiss**

오늘 배울 네이게이션의 **push ↔ pop**

### Navigation Controller 추가 방법



Navigation Controller를 추가하면 아래와 같이 Navigation Controller가 생김 (Navigation Controller를 Embed in한 뷰컨트롤러가 **Root**가 된다)



#### <이동 방법>

```
if let vc = storyboard?.instantiateViewController(withIdentifier: <#스토리보드 ID#>) {  
    navigationController?.pushViewController(secondVC, animated: true)  
}
```