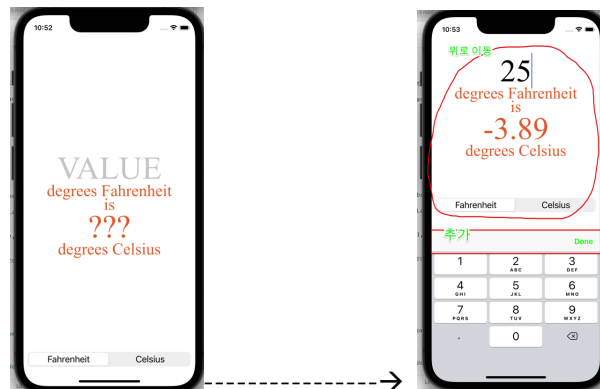


## Ch05 Lab

### 1. 다음과 같은 제어를 한번 해보자

- 키보드가 나오면 화면이 위로 이동하도록 변경한다
  - 이것은 "IS"에 또다른 constraint를 부착하여 2개의 constraints를 만든후 상황에 따라 필요한 한개를 적용한다.
  - SegmentControl의 하단 여백을 10  $\leftrightarrow$  키보드높이 만큼 상황에 따라 변경한다
- 키보드에 버튼을 부착하여 특별한 기능(여기서는 키보드 사라지게)을 부여한다



### 2. "키보드가 나오면 화면이 위로 이동하도록 변경한다"의 해결방안

- 다음과 같이 2개의 변수를 선언하라

```
class ConversionViewController: UIViewController {

    ...

    var isLabelYConstraintWithoutKeyboard: NSLayoutConstraint?
    var isLabelYConstraintWithKeyboard: NSLayoutConstraint?

    ...

}
```

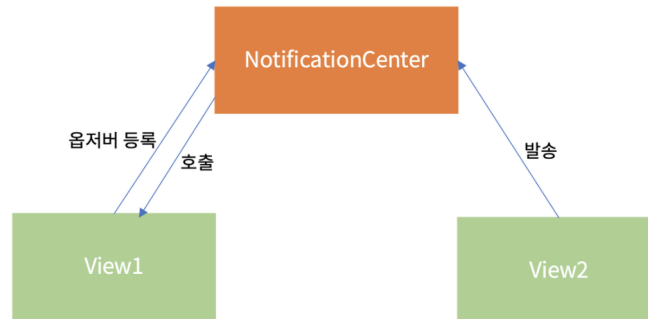
- viewDidLoad()함수에서 다음과 같이 코드를 변경하라

- isLabel에 두개의 constraint를 적용한다.

변경 전	<pre>isLabel.centerYAnchor.constraint(equalTo: view.safeAreaLayoutGuide.centerYAnchor).isActive = true</pre>
변경 후	<pre>isLabelYConstraintWithoutKeyboard = isLabel.centerYAnchor.constraint(equalTo: view.safeAreaLayoutGuide.centerYAnchor) isLabelYConstraintWithoutKeyboard?.isActive = true  isLabelYConstraintWithKeyboard = isLabel.centerYAnchor.constraint(equalTo: view.safeAreaLayoutGuide.centerYAnchor, constant: -200) isLabelYConstraintWithKeyboard?.isActive = false</pre>

- NotificationCenter 이용하여 키보드의 출현을 인식하자.

#### ■ 원리



#### ■ NotificationCenter와 Keyboard

- ◆ Notification은 noti피케이션 종류에 따라 유일한 이름 (`keyboardWillShowNotification`, `keyboardWillHideNotification`, "jmlee")을 가지고 있다.
- ◆ 누군가(상기그림에서 View1)는 특정 이름의 noti피케이션이 오면 나의 함수를 호출해 달라고 옵저버등록을 한다.
- ◆ 누군가(상기그림에서 View2) 특정 이름으로 noti피케이션을 발송하면 NotificationCenter에서는 그 이름으로 등록된 옵저버의 함수를 호출한다.

- viewDidLoad()함수의 마지막 부분에 다음과 같이 옵저버등록을 한다.

```

// NotificationCenter에 옵저버 등록을 한다.
// 키보드가 나타나면 keyboardWillShow 함수를 호출한다
NotificationCenter.default.addObserver(self,selector: #selector(keyboardWillShow),
    name: UIResponder.keyboardWillShowNotification, object: nil
)
// 키보드가 사라지면 keyboardWillHide 함수를 호출한다
NotificationCenter.default.addObserver(self,selector: #selector(keyboardWillHide),
    name: UIResponder.keyboardWillHideNotification,object: nil
)
  
```

- keyBoardWillShow, keyboardWillHide함수를 다음과 같이 구현한다

```

extension ConversionViewController{

    @objc private func keyboardWillShow(_ notification: Notification) {
        // 이것은 keyboardWillShow가 두번 연속호출되는 경우 그냥 리턴
        if isLabelYConstraintWithKeyboard?.isActive == true{
            return
        }
        isLabelYConstraintWithoutKeyboard?.isActive = false
        isLabelYConstraintWithKeyboard?.isActive = true
    }
}
  
```

```

@objc private func keyboardWillHide(_ notification: Notification) {
    isLabelYConstraintWithoutKeyboard?.isActive = true
    isLabelYConstraintWithKeyboard?.isActive = false
}
}

```

- 실행하여 보라

### 3. "segmentcontrol" 움직이기 해결방안

- 다음변수를 추가하라

```

class ConversionViewController: UIViewController {

    ...

    var isLabelYConstraintWithoutKeyboard: NSLayoutConstraint?
    var isLabelYConstraintWithKeyboard: NSLayoutConstraint?
    var segmentedControlBottomConstraint: NSLayoutConstraint?

    ...

}

```

- 다음과 같이 코드를 변경하라

변경전	segmentedControl.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant: -10).isActive = true	
변경후	segmentedControlBottomConstraint segmentedControl.bottomAnchor.constraint(equalTo: view.safeAreaLayoutGuide.bottomAnchor, constant: -10)  segmentedControlBottomConstraint?.isActive = true	=

- keyBoardWillShow, keyboardWillHide함수를 다음과 같이 변경하라

```

extension ConversionViewController{

    @objc private func keyboardWillShow(_ notification: Notification) {
        // 이것은 keyboardWillShow가 두번 연속호출되는 경우 그냥 리턴
        if isLabelYConstraintWithKeyboard?.isActive == true{
            return
        }
        isLabelYConstraintWithoutKeyboard?.isActive = false
        isLabelYConstraintWithKeyboard?.isActive = true
        segmentedControlBottomConstraint?.constant -= 350
    }
}

```

```

}

@objc private func keyboardWillHide(_ notification: Notification) {

    isLabelYConstraintWithoutKeyboard?.isActive = true
    isLabelYConstraintWithKeyboard?.isActive = false
    segmentedControlBottomConstraint?.constant += 350
}
}

```

- keyBoardWillShow, keyboardWillHide 함수를 다음과 같이 변경하라
  - ◆ 잘 동작할 것이다.
  - ◆ isAlive 대신에 priority를 이용해서도 가능하다. 한번 해 보길...

#### 4. Landscape에서도 잘되는가?

- 잘 안된다.
- 그래서 landscape에서는 아예 막아보자. 다음과 같이 코드를 변경해보라

```

extension ConversionViewController{

    @objc private func keyboardWillShow(_ notification: Notification) {
        // landscape에서는 아예 리턴하자
        if UIApplication.shared.statusBarOrientation.isLandscape {
            isLabelYConstraintWithoutKeyboard.isActive = true
            isLabelYConstraintWithKeyboard.isActive = false
            segmentedControlBottomConstraint.constant = 10
            return
        }
        // 이것은 keyboardWillShow가 두번 연속호출되는 경우 그냥 리턴
        ...
    }

    @objc private func keyboardWillHide(_ notification: Notification) {
        if UIApplication.shared.statusBarOrientation.isLandscape {
            return
        }
        isLabelYConstraintWithoutKeyboard?.isActive = true
        ...
    }
}

```

## 5. UITextField의 inputAccessoryView

- The default value of this property is nil. Assigning a view to this property causes that view to be displayed above the standard system keyboard (or above the custom input view if one is provided) when the text field becomes the first responder. For example, you could use this property to attach a custom toolbar to the keyboard.

- viewDidLoad()함수의 마지막에 다음 코드를 넣어라

```
let toolBarKeyboard = UIToolbar()
toolBarKeyboard.sizeToFit()
let btnDoneBar = UIBarButtonItem(title: "Done", style: .done, target: self, action:
#selector(self.dismissKeyboard))
let space = UIBarButtonItem(barButtonSystemItem: .flexibleSpace, target: nil, action: nil)
toolBarKeyboard.items = [space, btnDoneBar]
toolBarKeyboard.tintColor = .green

fahrenheitTextField.inputAccessoryView = toolBarKeyboard
```

- dismissKeyboard함수의 매개변수를 다음과 같이 고쳐라

```
extension ConversionViewController{
    @objc func dismissKeyboard(sender: Any){
        fahrenheitTextField.resignFirstResponder()
    }
}
```

- 동작해 보라

-