

2019 하계 ARGOS TEAM

---

오늘은  
Swift  
하는 날

2019.07.02  
Mac 써보실? 팀  
1회차

# CONTENTS

## 1. Mac 기본 사용법

- 단축키 위주 설명

## 2. 가상머신 및 Xcode 설치

- Vmware 설치 및 iso 파일 다운
- 버전 별 Xcode 설치하기

## 3. Xcode 살펴보기

- 구성 요소
- Story board
- 간단한 예제 화면 구성하기

## 4. Swift 문법 기초

- 이번 챕터의 정말 기본 문법 정리

## Mac 기본 사용법 기능별 단축키

## 알아 두면 좋은 단축키 12가지



## Using Mac OS

1. 여러 창을 한 번에 감추기: command + option + H
2. 밑의 프로그램 창(Dock)을 감추기: Command + option + D
3. 전체 화면 캡처 : command + shift + 3
4. 부분 화면 캡처: command + shift + 4 (+ space)<- 특정 화면 캡처
5. 사파리 읽기 목록 저장(페이지 임시 저장): command + shift + D
6. 작업 시 창과 창 사이 이동: Command + tab
7. 사용 탭 바로 닫기 : command + w
8. 삭제하기: command + delete
9. 한영 전환 단축키: control + space
10. 붙여넣기: command + v
11. 복사: command + c
12. 앱 강제 종료: option + command + esc

### ※ 윈도우 키보드일 경우 주의할 키

- Command = window Key
- Option = Alt
- Control = Ctrl

가상머신 설치  
Vmware/ iso File

아래 링크를 참조하여 하나씩 따라해보세요.  
문제가 생기면 팀장에게 질문해주세요

Vmware Mac OS X 설치 ▼

<https://jeongchul.tistory.com/607>

1. Vmware 다운로드
2. Mac OS X 이미지 파일 다운
3. Vmware unlock 다운 후 설치 옵션 생성
4. 이미지 파일로 가상머신에서 생성
5. Darwin.iso 파일로 vmware tool 설치
6. 설치 완료!

Mac을 보유하고 계시면 이 챕터는 패스해주세요!

Xcode 설치  
버전 별 설치 방법

Mac이 있으신 분들은 바로 App store에서 다운받아주세요.  
문제가 생기면 팀장에게 질문해주세요용

Xcode 설치 ▼

<https://myjamong.tistory.com/1>

- Mac은 최신버전 업데이트가 되었을 경우 문제없이 App store에서 'Xcode' 검색 후 다운 가능
- 가상머신 사용자는 최대 설치 버전이 **10.14 이므로 App store에서는 다운 불가능**
- 링크의 글을 참고하여 최소 8.3.3 버전을 다운 권장

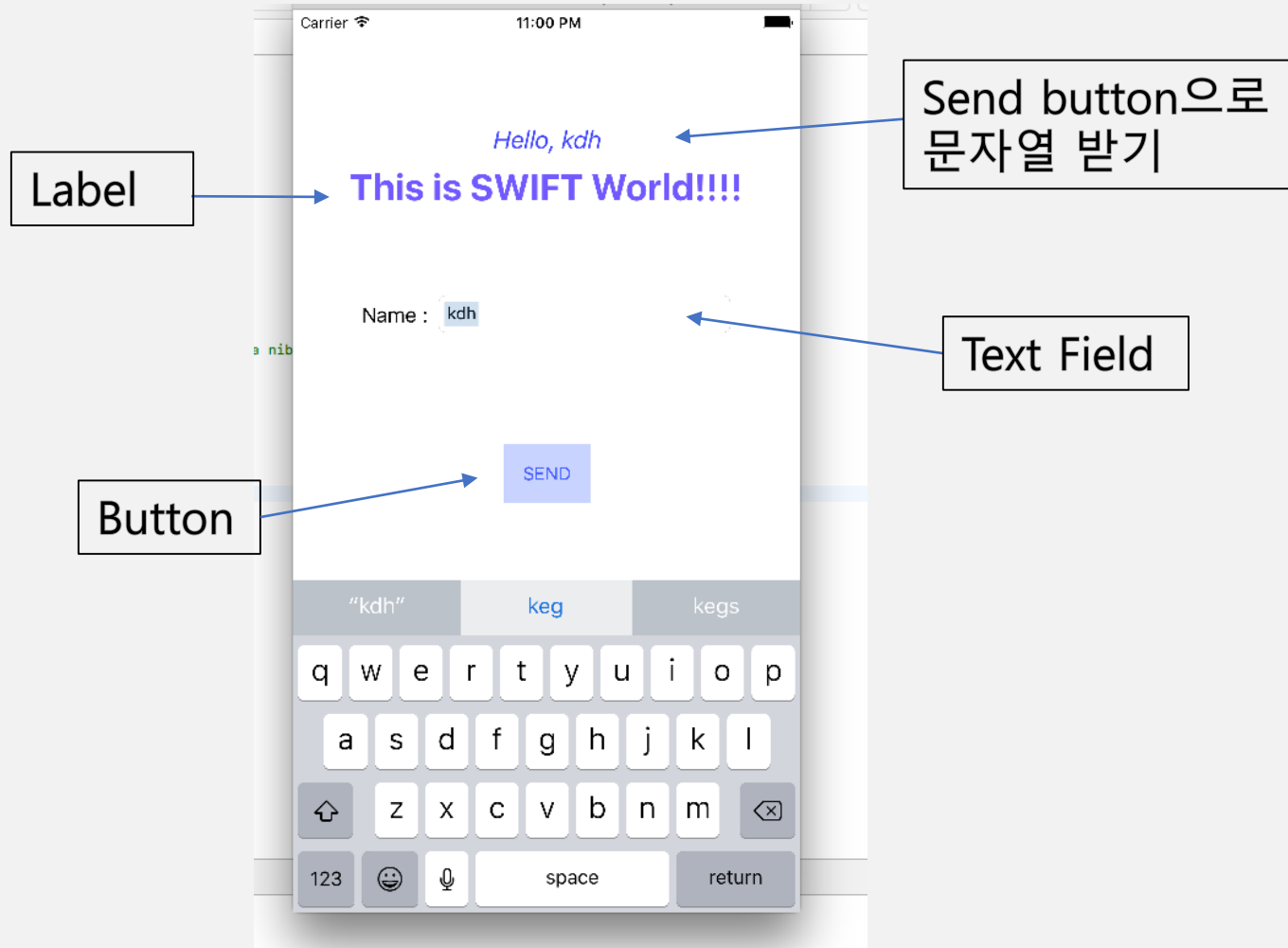
Mac을 보유하고 계시면 이 챕터는 패스해주세요!

## 오늘의 목표

Xcode 다루기

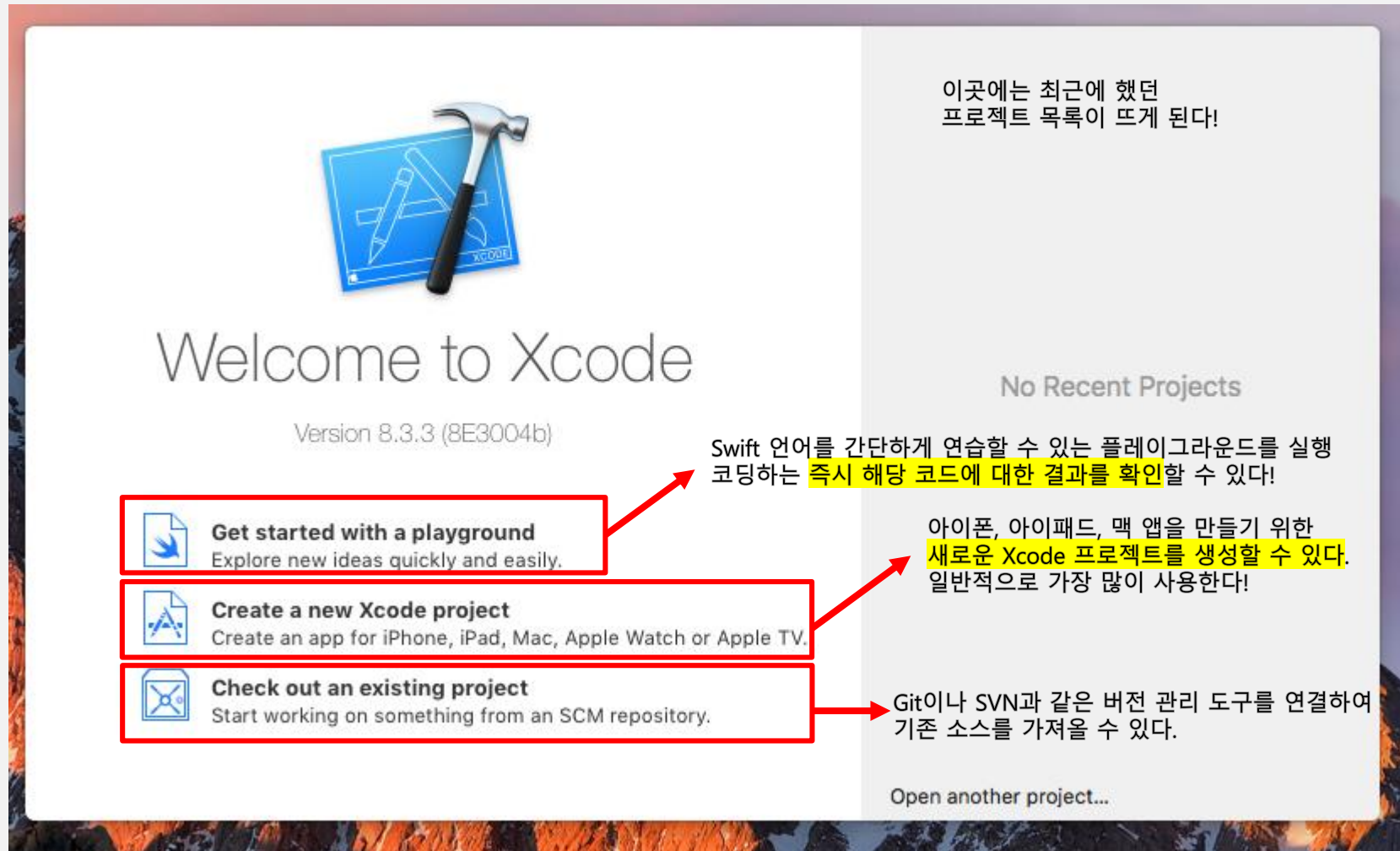
간단하게 오늘의 목표를 정해봅시다.

# 처음에는 역시 Hello world...



## Xcode 살펴 보기 구성 요소

시작하기 전에 어떤 툴인지 봅시다.



## Xcode 살펴 보기 구성 요소

시작하기 전에 어떤 툴인지 봅시다.

The diagram illustrates the project structure of 'FristTestApp' in Xcode. It shows a sidebar with a file explorer and a main workspace with a file list. An orange arrow points from the sidebar to the main workspace. The file list in the main workspace is numbered 1 through 6, with orange stars next to items 2, 3, and 4. A red-bordered box on the right contains a list of descriptions for these numbered items.

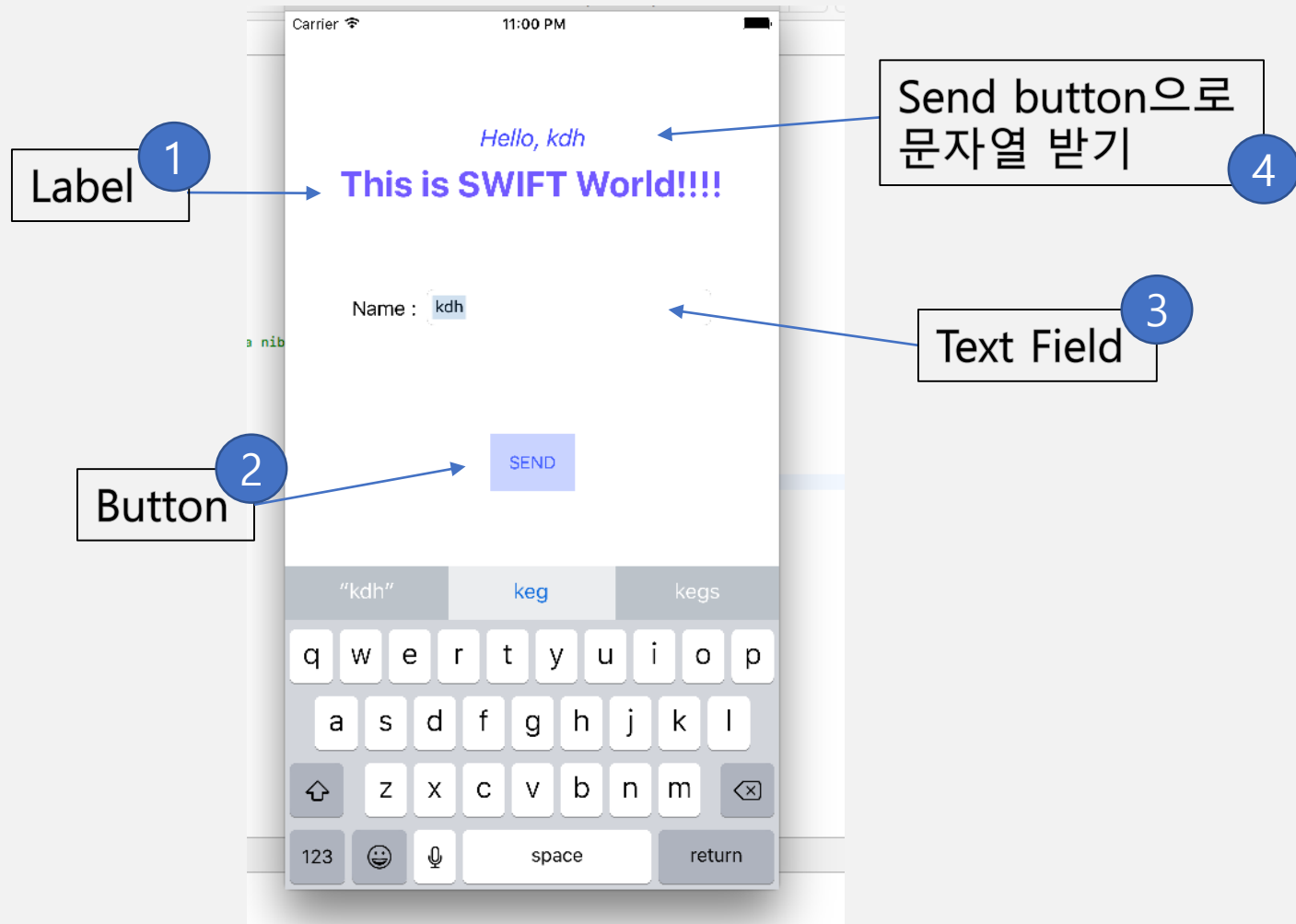
- (1) AppDelegate.swift
- (2) ViewController.swift ★
- (3) Main.storyboard ★
- (4) Assets.xcassets ★
- (5) LaunchScreen.storyboard
- (6) Info.plist

- 1. 앱을 실행하거나 종료, 또는 백그라운드를 실행하는 일 관리 파일
- 2. 화면에서 보이는 뷰에서 처리하는 **내용의 swift 코드를 담고 있는 클래스**
- 3. 앱의 내용을 **시각적**으로 쉽게 이해할 수 있도록 그림으로 표현한 파일
- 4. 앱의 아이콘을 보관하는 저장소
- 5. 앱이 실행될 때 잠시 나타나는 스플래시 화면을 만드는 스토리 보드
- 6. 앱이 실행되는데 필요한 정보를 저장하고 있는 파일



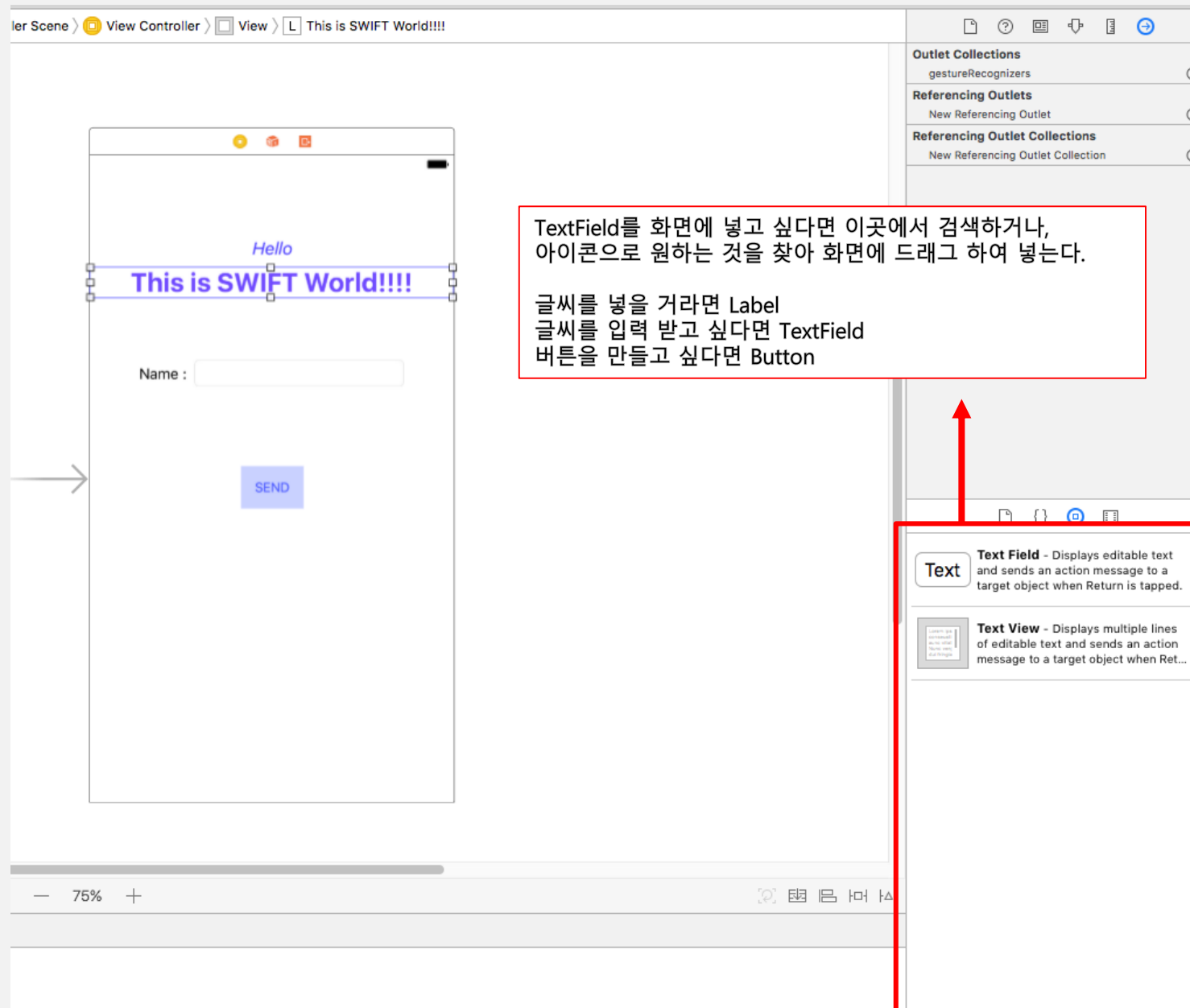
간단한 예제 화면  
구성하기

이제 간단한 App 화면을 구성해봅시다!



## 간단한 예제 화면 구성하기

## Label/Button/Text Field 화면에 구성하기



## 간단한 예제 화면 구성하기

## 아웃렛 변수 추가하기

The screenshot shows the Xcode interface for a project named 'FristTestApp'. The left pane displays the Storyboard with a UI containing a label 'Hello', a large label 'This is SWIFT World!!!!', and a text field labeled 'Name :'. A blue button labeled 'SEND' is at the bottom. A red arrow points from the 'SEND' button to the code editor, with the text 'Control 누르며 드래그하기' (Press Control and drag) written in red. The right pane shows the Swift code for 'ViewController.swift'. Two sections of code are highlighted with red boxes: the IBOutlet declarations and the btnSend action method.

```
// ViewController.swift
// FristTestApp
//
// Created by KDHPC on 2019. 6. 30..
// Copyright © 2019년 KDHPC. All rights reserved.

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var IbHello: UILabel!
    @IBOutlet weak var txtName: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view from the nib.
    }

    @IBAction func btnSend(_ sender: UIButton) {

    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

**<Mission>**

- Label에 대한 아웃렛 변수 추가하기
- TextField에 대한 변수 추가하기
- 버튼의 동작에 대한 코드 생성하기

모르면 검색해서 해봅시다

(+) Outlet 변수와 Action 메소드명  
수정 및 삭제  
검색해보기 꼭!!!

간단한 예제 화면  
구성하기

## 버튼 동작 구현하기

아래의 코드를 따라 치고  
애물레이터를 가동시켜  
우선 결과를 봅시다!

```
//
// ViewController.swift
// FristTestApp
//
// Created by KDHPC on 2019. 6. 30..
// Copyright © 2019년 KDHPC. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var IbHello: UILabel!
    @IBOutlet weak var txtName: UITextField!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a
        nib.
    }

    @IBAction func btnSend( sender: UIButton) {
        IbHello.text = "Hello, " + txtName.text!
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

"Hello, "라는 문자열과  
txtName text의 문자열을 결합하여  
lbHello text에 넣는다 라는 의미

```
1      2  
@IBAction func btnSend(_ sender: UIButton) {  
    3      4  
    lbHello.text = "Hello, " + txtName.text!  
}
```

1. 함수를 선언할 때 **func 키워드**를 사용해서 선언한다.
2. 액션 함수가 실행되도록 이벤트를 보내는 객체  
(즉 여기서는 버튼 객체에서 이벤트가 발생했을 때 해당 액션 함수를 실행시킬 것이므로 UIButton 클래스 타입을 선택한다.)
3. 레이블 객체가 가지고 있는 속성 중 텍스트의 내용을 의미하는 속성
4. 텍스트필드 객체가 가지고 있는 속성 중 텍스트 내용을 의미하는 속성

Swift 문법 기초  
정말 기초

스위프트도 아직까지는 정말 별거 없었습니다..  
앞으로도 열심히!

Swift는 문장 끝에 ;가 없다 ▼

다른 언어와 비슷하면서도 다르다!

Swift는 문장 끝에 ;를 붙이지 않아도 되므로 편리하다!  
하지만 ;를 붙여도 에러가 발생하지는 않는다

상수와 변수, 함수, 메서드 이름은? ▼

일반 언어들과 이름 지정 시 따르는 규칙과 같다!

1. 유니코드 포함 어떤 문자이던지 사용 가능
2. 숫자로 시작할 수 없음
3. 예약어나 키워드 등록된 단어는 사용 불가능

마치며..  
다음부터는

오늘 열심히 따라와 주셔서 감사합니다.  
앞으로도 같이 열심히 해봐요!

저도 지금 조금씩 배우는 입장이라 아직 많이 미숙합니다!

그래도 일단 모르는 문제가 발생하면 같이 톡방이나 모임 시간에 공유하고 같이 공부하도록 합시다!

다음 차시 팀모임부터는 제가 먼저 만든 앱 화면만을 보여주고  
여러분들이 똑같이 찾아서 만들어가는 형식으로 진행하려고 합니다.

이제부터는 제가 알려주지 않습니다!

저는 문제만 제시하고 여러분들이 찾아서 풀어 헤쳐 나가야 해요 ㅎㅎ

하지만 모르는 것이 있으면 언제든지 물어봐 주세요

여름방학 팀 모임 다같이 파이팅!!!

**수고하셨습니다!**

다음 팀 모임

7월 9일 화요일

오후 1시~3시