

08 장. 맵 뷰로 지도 나타내기

이번 장에서는 지도, 위치 관련 앱에서 활용할 수 있는 맵 뷰(Map View)를 마스터해 보자.

여기서는 맵 뷰를 이용해 위도와 경도 그리고 범위를 설정하여 지도에 나타내고, 원하는 곳에 핀을 설치하여 원하는 글자를 나타낼 것이다.

위치를 선택하기 위해 버튼 대신 '세그먼트 컨트롤'을 이용한다. 세그먼트란 세분화된 기능을 가진 버튼을 말하며, 세그먼트 컨트롤은 여러개의 세그먼트를 수평으로 나열하여 구성된 수평 컨트롤을 지정한다, 이러한 세그먼트를 세 개 만들어 각각의 위치를 표시한 다음 '폴리텍대학'과 '이지스 퍼블리싱 출판사'에 핀을 설치해 보겠다.



08-1 맵 뷰란?

사람들은 평소 길을 찾을 때 내비게이션 역할을 해주는 지도 앱을 많이 사용한다. 기본적으로 지도 화면에 현재 내 위치를 알려줄 뿐만 아니라 이동도 가능한 기능이 있다.

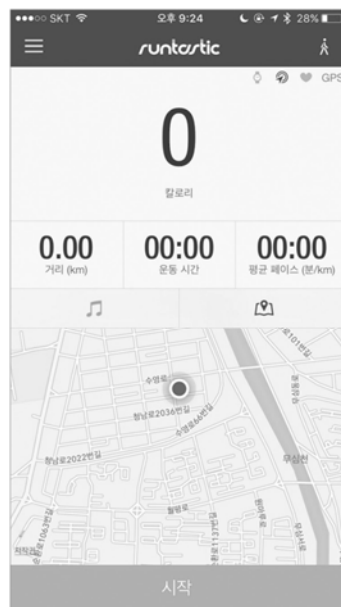
구글 지도, 네이버 지도, 다음 지도 외에 '런타스틱(Runtastic)' 같은 운동 관련 앱, '서울 버스' 앱과 같은 버스 앱이 그 예이다.

이런 앱은 '맵 뷰'를 이용하여 쉽게 만들 수 있다. 사용자의 경도, 위도 및 고도와 같은 위치 정보를 이용하여 사용자의 위치를 지도에 표시하고 추적할 수 있다.

또한 특정 위치를 표시하고 사용자의 터치를 인식하여 확대, 축소 및 이동 기능을 제공한다.



아이폰의 기본 지도 앱



런타스틱(Runtastic) 앱



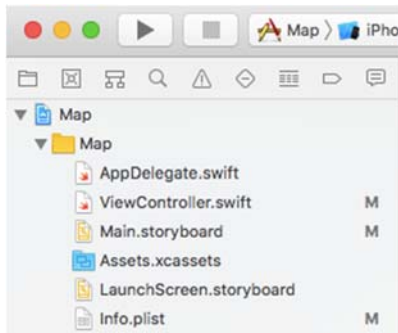
청주 버스 앱

08-2 맵뷰앱을위한기본환경구성하기

편집기 영역의 뷰 컨트롤러(View Controller)를 보면 정사각형 모양으로 되어 있다. 정사각형 모양의 뷰 컨트롤러는 기기의 방향에 따라 자유자재로 레이아웃을 구현하는 기능인 자동 레이아웃(Auto Layout)이 가능한 뷰 컨트롤러이다.

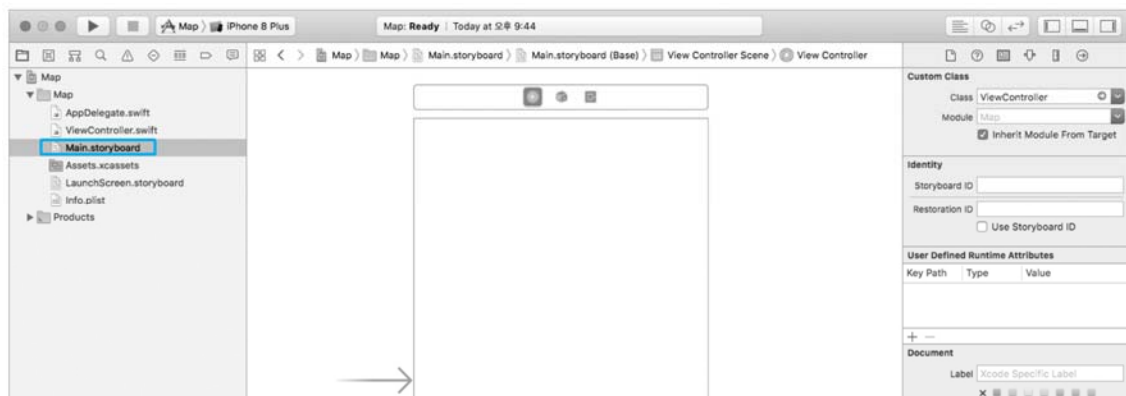
그러나 이번 프로젝트에서는 자동 레이아웃 기능을 사용하지 않고 아이폰 크기로 고정된 채 앱을 구현할 것이므로 뷰 컨트롤러의 모양을 아이폰 형태로 바꿔보자.

1. Xcode 를 실행한 후 [Create a new Xcode project]를 클릭하여 새 프로젝트를 만든다. 프로젝트의 이름은 'Map'으로 입력한다.



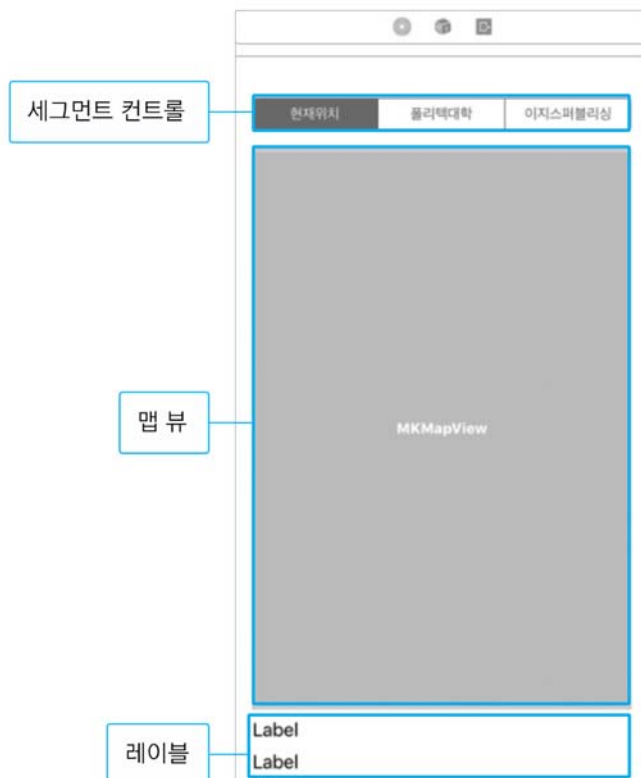
2. 뷰 컨트롤러 크기 조절하기

아이폰 모양의 뷰 컨트롤러 크기를 상황에 맞게 조절한다.



08-3 스토리보드로 맵뷰 앱 화면 꾸미기

이 앱에서는 현재 위치와 폴리텍대학 및 이지스퍼블리싱 출판사의 위치를 지도에 표시해 보자. 지도를 보여 주기 위한 맵 뷰(Map View)객체, 세 곳의 위치를 선택하기 위한 세그먼트 컨트롤(Segmented Control)객체 그리고 위치 정보를 표시하기 위한 레이블(Label)객체를 사용한다. 오른쪽 그림은 완성된 스토리보드 화면으로 이 그림과 사용된 객체 목록을 참고하여 배치해 보자.



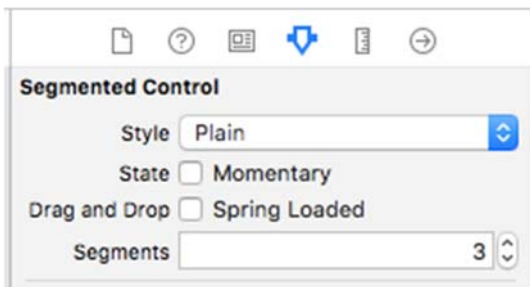
1. 세그먼트 컨트롤 추가하기

지도 화면에서 현재 위치 및 특정 위치를 선택할 수 있는 '세그먼트 컨트롤(Segmented Control)'을 만들어 추가해 보자. 먼저 화면 오른쪽 아랫부분의 오브젝트 라이브러리의 검색란에 'se'라고 입력한 후 [세그먼트 컨트롤(Segmented Control)]을 찾아 스토리보드 화면으로 끌어와 다음 그림과 같이 위치에 배치한다. 그리고 좌우로 늘려준다.

2. 세그먼트 컨트롤에 들어갈 글자 변경하기

이 세그먼트 컨트롤을 누르면 특정 위치를 선택할 수 있도록 설정하겠다.

[세그먼트 컨트롤(Segmented Control)]을 선택한 채 오른쪽의 인스펙터 영역에서 [Attributes inspector]버튼을 클릭한 후 Segments 를 [3]으로 선택해 세그먼트를 하나 추가한다.



3. 마지막 세그먼트에 타이틀을 추가하기 위해 Segment 에서 [Segment 2]를 선택한다.

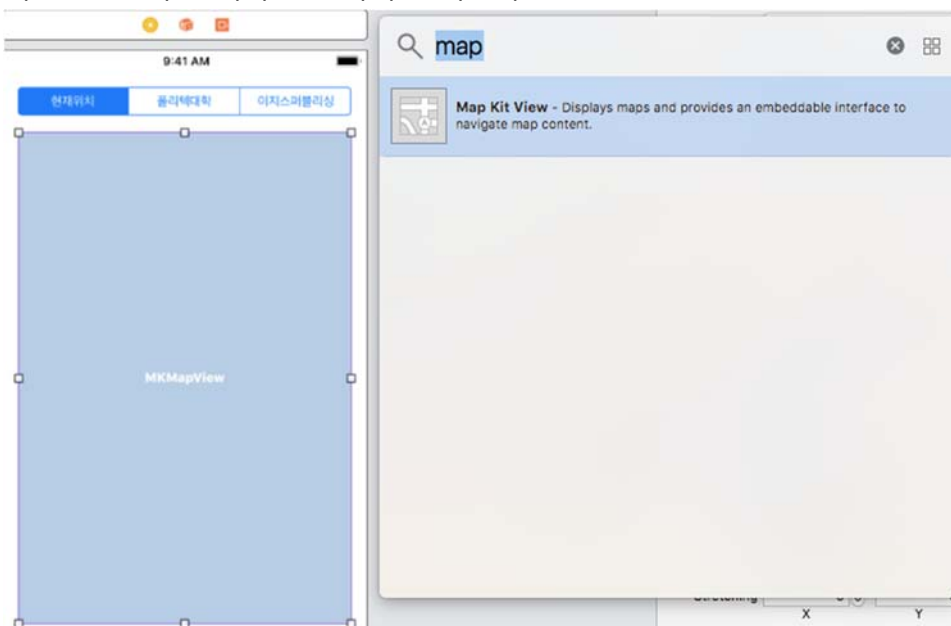
4. 'Title'입력란에 지도에서 찾고자 하는 장소의 이름을 입력한다. 필자의 경우는 출판사의 위치를 선택하기 위해 '이지스퍼블리싱'을 입력한다.



6. 지도를 보여 줄 맵 뷰 추가하기

세그먼트 컨트롤에 대해 살펴보았으니 이제는 지도를 보여 줄 맵 뷰를 추가해 보자. 화면 오른쪽 아랫 부분의 오브젝트 라이브러리의 검색란에 'ma'라고 입력하여 [맵 키트 뷰 (Map Kit View)]라는 객체를 찾는다.

맵 뷰를 스토리보드로 끌어온 후 세그먼트 컨트롤 아래에 배치한다. 이때 아래쪽은 위치 정보를 위한 레이블을 위해 남겨둔다.



7. 위치 정보를 표기할 레이블 추가하기

맵 뷰아래쪽에는 그 위치에 대한 정보를 표기해 보자. 정보를 보여 주기 위해 여기서는 [레이블(Label)]을 이용한다. 오른쪽 아랫부분의 오브젝트 라이브러리의 검색란에 'la'를 입력하면 목록에 [레이블(Label)]이 나타난다. 레이블 두 개를 스토리보드로 끌어와 맵 뷰아래에 배치한 후 크기도 조절한다. 하나에는 위치에 대한 설명을 나머지 하나에는 실제 위치의 주소를 표기할 것이다.



08-4 아웃렛 변수와 액션 함수 추가하기

프로그램에서 지도를 보여 주기 위한 맵 뷰의 아웃렛 변수와 위치 정보를 보여 주기 위한 레이블의 아웃렛 변수 그리고 미리 지정한 위치를 지도에 보여 주기 위한 세그먼트 컨트롤의 액션 함수를 추가해 보겠다.

1. 보조 편집기 영역 열기

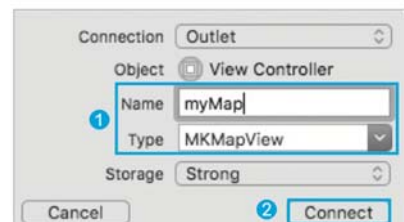
아웃렛 변수와 액션 함수를 추가하기 위해서는 우선 화면 오른쪽 윗부분의 [Show the Assistant editor] 버튼을 클릭하여 보조 편집기 영역(Assistant editor)을 열어야 한다. 그러면 가운데 화면의 스토리보드 부분이 두 영역으로 나누어지면서 왼쪽에는 스토리보드, 오른쪽에는 보조 편집기 영역이 나타난다.

2. 맵 뷰에 대한 아웃렛 변수 추가하기

맵 뷰를 마우스 오른쪽 버튼으로 클릭한 후 오른쪽 보조 편집기 영역으로 드래그하면 다음 그림과 같이 연결선이 나타난다. 드래그한 연결선을 뷰 컨트롤러의 클래스 (ViewController class) 선언문 바로 아래에 연결하여 아웃렛 변수를 추가한다.

3. 아웃렛 변수 설정 창이 나타나면 아래 표를 참조하여 아웃렛 변수의 이름을 입력한 후 [Connect] 버튼을 클릭한다, 여기서는 이름(Name)을 'myMap'으로 입력한다.

위치	뷰 컨트롤러 클래스 선언문 바로 아래
연결(Connection)	Outlet
이름(Name)	myMap
유형(Type)	MKMapView



```
@IBOutlet var myMap: MKMapView!
```

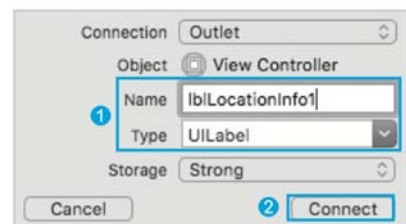
4. 그림에서 위치를 참고하여 코드에 'import MapKit'를 추가한다. 이 맵 키트(MapKit)는 지도를 확대, 축소 및 이동하는 등 지도에 관한 여러 기능을 제공한다. 따라서 사용자의 터치를 인식하여 기능을 수행하기 위해선 이 작업이 필요하다.



5. 다시 레이블을 마우스 오른쪽 버튼으로 클릭한 후 오른쪽 보조 편집기 영역으로 드래그하면 다음 그림과 같이 연결선이 나타난다. 드래그한 연결선을 위해서 추가한 아웃렛 변수의 바로 아래에 배치한 후 마우스 버튼에서 손을 떼는다.

6. 설정 창이 나타나면 다음 표를 참고하여 아웃렛 변수의 이름을 입력한 후 [Connect] 버튼을 클릭하여 레이블과 아웃렛 변수를 연결한다.

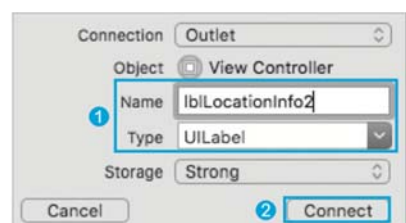
위치	myMap 아웃렛 변수 바로 아래
연결(Connection)	Outlet
이름(Name)	lblLocationInfo1
유형(Type)	UILabel



```
@IBOutlet var lblLocationInfo1: UILabel!
```

7. 같은 방법으로 두 번째 레이블의 설정 창에서 아웃렛 변수의 이름을 입력한 후 [Connect] 버튼을 클릭하여 레이블과 아웃렛 변수를 연결한다. 여기서 이름을 'lblLocationInfo2'로 입력한다.

위치	lblLocationInfo1 아웃렛 변수 바로 아래
연결(Connection)	Outlet
이름(Name)	lblLocationInfo2
유형(Type)	UILabel



```
@IBOutlet var lblLocationInfo2: UILabel!
```

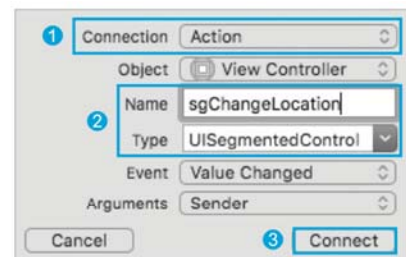

8. 액션 함수 추가하기

세그먼트 컨트롤에 대한 액션 함수를 추가하자.

마우스 오른쪽 버튼으로 세그먼트를 클릭한 후 드래그해서 오른쪽 보조 편집기 영역에 갖다 놓는다.

아래표를 참조하여 세그먼트 컨트롤의 액션 함수를 연결한다. 이때 유형(Type)을 [UISegmentedControl]로 설정하는 이유는 세그먼트 컨트롤에 액션을 추가하기 위해서이다.

위치	뷰 컨트롤러 클래스의 마지막 '}' 바로 앞 부분
연결(Connection)	Action
이름(Name)	sgChangeLocation
유형(Type)	UISegmentedControl



```
@IBAction func sgChangeLocation(_ sender: UISegmentedControl) {  
}
```

08-5 지도보여주기

지도를 보여주기 위한 변수 선언과 초기 작업을 진행해 보자.

1. 스펀더드 에디터로 화면 모드 수정하기

왼쪽 네비게이터에서 [ViewController.swift]를 선택한다.

2. 지도 보여주기

지도를 보여주기 위해 변수와 델리게이트를 선언해 준다.

아래 그림을 참고하여 다음 소스를 뷰 컨트롤러의 선언문과 아웃렛 변수들 아래에 추가한다.

```
Class ViewController: UIViewController, CLLocationManagerDelegate {  
    :  
    let locationManager = CLLocationManager()  
    :  
}
```

3. 앱을 실행하면 지도가 나타나도록 viewDidLoad 함수에 코드를 추가한다. 다음 그림을 참고하여 소스를 입력한다.

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    // Do any additional setup after loading the view, typically from a nib.  
    lblLocationInfo1.text = ""  
    lblLocationInfo2.text = ""  
    locationManager.delegate = self  
    locationManager.desiredAccuracy = kCLLocationAccuracyBest  
    locationManager.requestWhenInUseAuthorization()  
    locationManager.startUpdatingLocation()  
    myMap.showsUserLocation = true  
}
```

- 1) 위치 정보를 표시할 레이블에는 아직 특별히 표시할 필요가 없다.
- 2) 상수 locationManager의 델리게이트를 self로 설정한다.
- 3) 정확도를 최고로 설정한다.
- 4) 위치 데이터를 추적하기 위해 사용자에게 승인을 요구한다.
- 5) 위치 업데이트를 시작한다.
- 6) 위치보기 값을 true로 설정한다.

4. 결과 보기

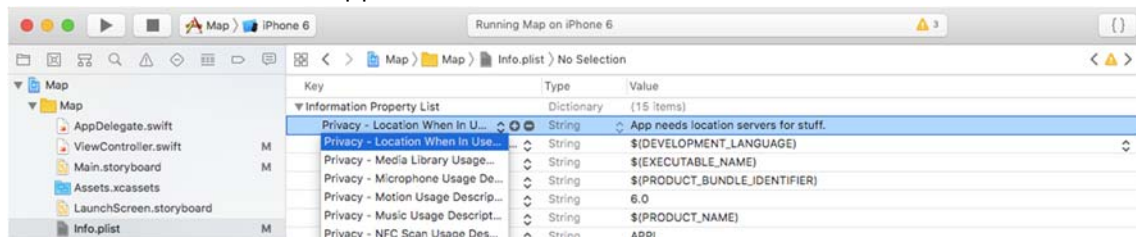
이제 지도가 제대로 나타나는지 확인해 보겠다. 실행을 하기전에 iOS 시뮬레이터의 기기를 원하는 기기로 변경한다. [실행] 버튼을 클릭하면 iOS 시뮬레이터가 구동되며 프로그램한 앱이 실행된다, 우리나라 지도가 나타난다.

5. 앱이 사용자 위치에 접근하도록 허용하기

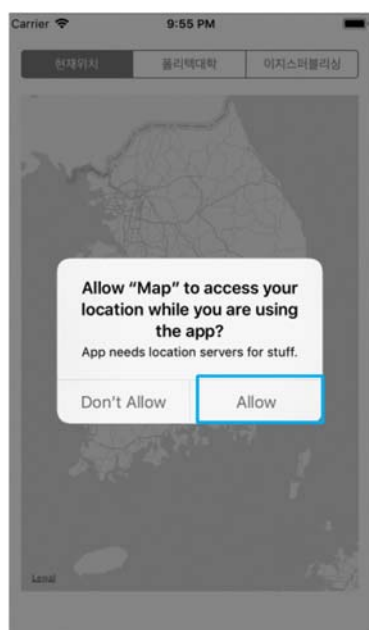
경우에 따라 지도가 나타나지 않을 수도 있다. 이런 경우 경고창을 확인하여 'This app has attempted to access privacy-sensitive data without a usage description' 이라는 메시지가 있다면 'info.plist'을 수정하여 해결할 수 있다.

'info.plist'를 수정하기 위해 왼쪽의 내비게이터 영역에서 [info.plist]를 선택한 후 마우스의 커서를 [information Property List]위로 가져가 [+]가 표시되면 이를 클릭한다.

리스트가 새로 추가되면 'Privacy-Location When In Use Usage Description'을 선택한다. [Value]를 더블클릭하여 [App needs location servers for stuff.]로 수정한다.



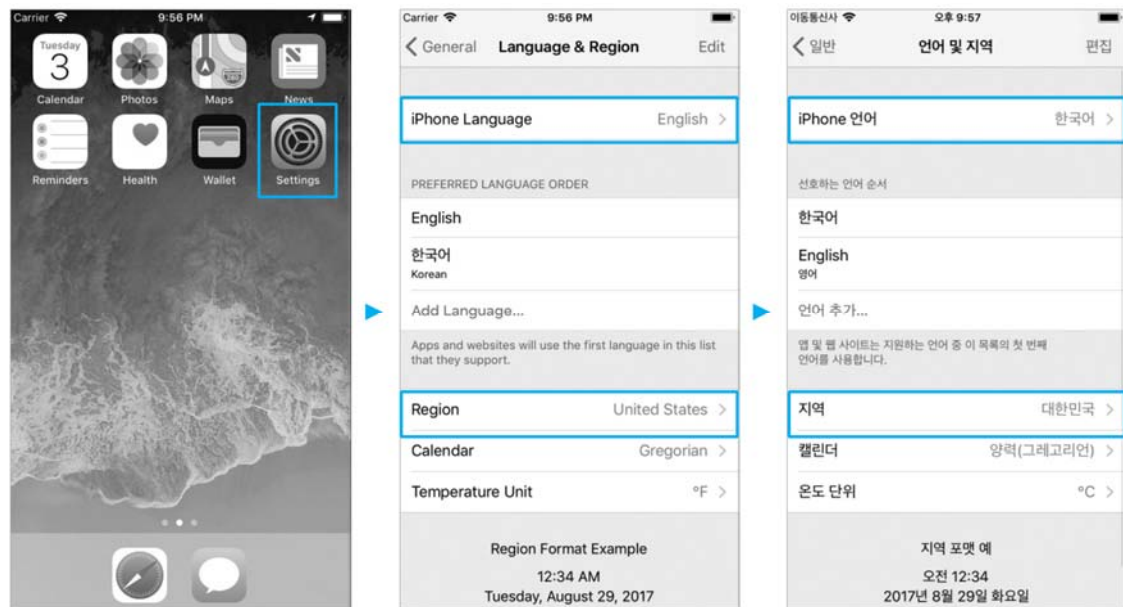
6. 다시 시뮬레이터를 실행해 앱이 사용자 위치에 접근하도록 허용할 것인지를 물어보면 [Allow]를 눌러서 허용한다. 다음부터 시뮬레이터를 실행하더라도 경고 메시지가 나타나지 않는다.



08-6 시뮬레이터를 한글화하기

지도를 보면 '대한민국'이 아닌 'SOUTH KOREA'로 표시되어 있는데, 이는 시뮬레이터의 언어가 영어로 되어 있기 때문이다. 설정을 통해 언어와 지역을 수정해 보자.

1. 먼저 iOS 시뮬레이터에서 홈 버튼을 클릭하거나 키보드의 command 키 + shift 키 + H 키를 눌러 홈으로 이동한다. 그런 다음 마우스를 이용해 [Settings -> General -> Language & Region]을 클릭하여 이동한다. 현재 언어는 [English]로, 위치는 [United States]로 되어 있는데, 이를 [한국어]와 [대한민국]으로 수정한다.



2. 다시 홈 버튼을 클릭하거나 키보드의 command 키 + shift 키 + H 키를 눌러 홈으로 이동하면 아이폰의 글자들이 한글로 변경된 것을 확인할 수 있다.



08-7 위도와 경도로 원하는 위치 표시하기

앞에서는 우리나라의 전체 지도를 보여 주었다면 이제는 사용자가 원하는 위도와 경도의 지도만 보여 주기 위해 추가적인 코딩을 해본다.

1. 추가하는 함수 이름을 'goLocation'으로 지정하고, 입력 파라미터는 위도 값, 경도 값, 범위로 설정한다. 이 함수를 앞에서 만든 `viewDidLoad` 액션#함수#바로#앞에#입력한다#

5# 지도를# 나타내기# 위해서는# 네# 개의# 함수를# 호출해야# 한다# 그# 네가지# 함수는#
`CLLocationCoordinate2DMake` 함수# `MKCoordinateSpanMake` 함수#
`MKCoordinateRegionMake` 함수#그리고#맵#뷰의#`setRegion` 함수이다1

```
func goLocation(latitudeValue: CLLocationDegrees,  
    longitudeValue : CLLocationDegrees, delta span :Double) {  
    let pLocation = CLLocationCoordinate2DMake(latitudeValue, longitudeValue) —①  
    let spanValue = MKCoordinateSpanMake(span, span) —②  
    let pRegion = MKCoordinateRegionMake(pLocation, spanValue) —③  
    myMap.setRegion(pRegion, animated: true) —④  
}
```

1) 위도 값과 경도 값을 매개변수로 하여 `CLLocationCoordinate2DMake` 함수를 호출하고, 리턴 값을 `pLocation` 으로 받는다

2) 범위 값을 매개변수로 `MKCoordinateSpanMake` 함수를 호출하고, 리턴 값을 `spanValue` 로 받는다.

3) `pLocation` 과 `spanValue` 값을 매개변수로 하여 `MKCoordinateRegionMake` 함수를 호출하고, 리턴 값을 `pRegion` 으로 받는다.

4) `pRegion` 값을 매개변수로 하여 `myMap.setRegion` 함수를 호출한다.

3. 위치가 업데이트되었을 때 지도에 위치를 나타내기 위해 함수를 추가하자.

`Location Manager` 함수를 2 번 과정에서 추가한 `goLocation` 함수 아래에 추가한다.

다음 소스를 `locationManager` 함수안에 입력하면 마지막 위치가 지도에 나타난다.

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations  
    locations: [CLLocation]) {  
    let pLocation = locations.last —①  
    goLocation(latitudeValue: (pLocation?.coordinate.latitude)!,  
        longitudeValue: (pLocation?.coordinate.longitude)!, delta: 0.01) —②  
}
```

4. 다음 소스를 locationManager 함수 안에 입력하면 마지막 위치가 지도에 나타난다.

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations
    locations: [CLLocation]) {
    let pLocation = locations.last —❶
    goLocation(latitudeValue: (pLocation?.coordinate.latitude)!,
        longitudeValue: (pLocation?.coordinate.longitude)!, delta: 0.01) —❷
}
```

- 1) 위치가 업데이트되면 먼저 마지막 위치 값을 찾아낸다.
 - 2) 마지막 위치의 위도와 경도값을 가지고 앞에서 만든 goLocation 함수를 호출한다.
- 이때 delta 값은 지도의 크기를 정하는데, 값이 적을 수록 확대되는 효과가 나타난다. delta 를 0.01 로 하였으니 1 의 값보다 지도를 100 배로 확대해서 보여 줄 것이다.

08-8 위치 정보 추출해 텍스트로 표시하기

이제 위도와 경도 값을 이용해 정보를 가져오고 나라, 지역 및 도로명을 찾아 레이블에 표시해 보자.

1. 위도와 경도 값을 가지고 역으로 위치의 정보, 즉 주소를 찾아보자.
핸들러의 익명 함수를 추가로 준비한다.
2. 핸들러의 추가 함수를 작성하는데 placemarks 값의 첫번 부분만 pm 상수로 받는다.
이렇게 pm 상수에서 나라(country)을 추출하고, 지역(locality)과 도로(throughfare)는 존재할 경우에 출력하도록 하자. 각각의 값 사이에는 공백을 넣어 읽기 쉽게 한다.
마지막으로 레이블에 위치 정보를 넣는다.

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations
    locations: [CLLocation]) {
    let pLocation = locations.last
    goLocation(latitudeValue: (pLocation?.coordinate.latitude)!,
        longitudeValue: (pLocation?.coordinate.longitude)!, delta: 0.01)
    CLGeocoder().reverseGeocodeLocation(pLocation!, completionHandler: {
        (placemarks, error) -> Void in
        let pm = placemarks!.first —①
        let country = pm!.country —②
        var address:String = country! —③
        if pm!.locality != nil { —④
            address += " "
            address += pm!.locality!
        }
        if pm!.thoroughfare != nil { —⑤
            address += " "
            address += pm!.thoroughfare!
        }

        self.lblLocationInfo1.text = "현재 위치" —⑥
        self.lblLocationInfo2.text = address —⑦

    })

    locationManager.stopUpdatingLocation() —⑧
}
```

- 1) placemarks 값의 첫부분만 pm 상수로 대입한다.
- 2) pm 상수에서 나라 값을 country 상수의 값을 대입한다.
- 3) 문자열 address 에 country 상수의 값을 대입한다.
- 4) pm 상수에서 지역 값이 존재하면 address 문자열에 추가한다.
- 5) pm 상수에서 도로 값이 존재하면 address 문자열에 추가한다.
- 6) 레이블에 "현재위치"텍스트를 표시한다.
- 7) 레이블에 address 문자열의 값을 표시한다.
- 8) 마지막으로 위치가 업데이트되는 것을 멈추게 한다.

스위프트 문법] nil 이란?

스위프트에서 nil 은 값이 존재하지 않음을 의미

아래 flightCode 중 flightNumber "aa"는 존재하지 않으므로 결과로 "항공사 코드 aa 는 nil 입니다."를 출력

즉, "aa"의 항공사 코드는 와 같이 나열한 존재하지 않으므로 nil 을 표시

```
var flightCode = [
    "oz": "아시아나항공",
    "ke": "대한항공",
    "ze": "이스타항공",
    "lj": "진에어",
    "7c": "제주항공",
]

flightNumber = "aa"
print("항공사 코드 \(flightNumber)는 \(flightCode[flightNumber])입니다.")
```

항공사 코드가 존재하는지를 if 문과 nil 을 활용해서 코드를 작성할 수 있음.
결과는 "없는 항공사 코드 입니다."를 출력

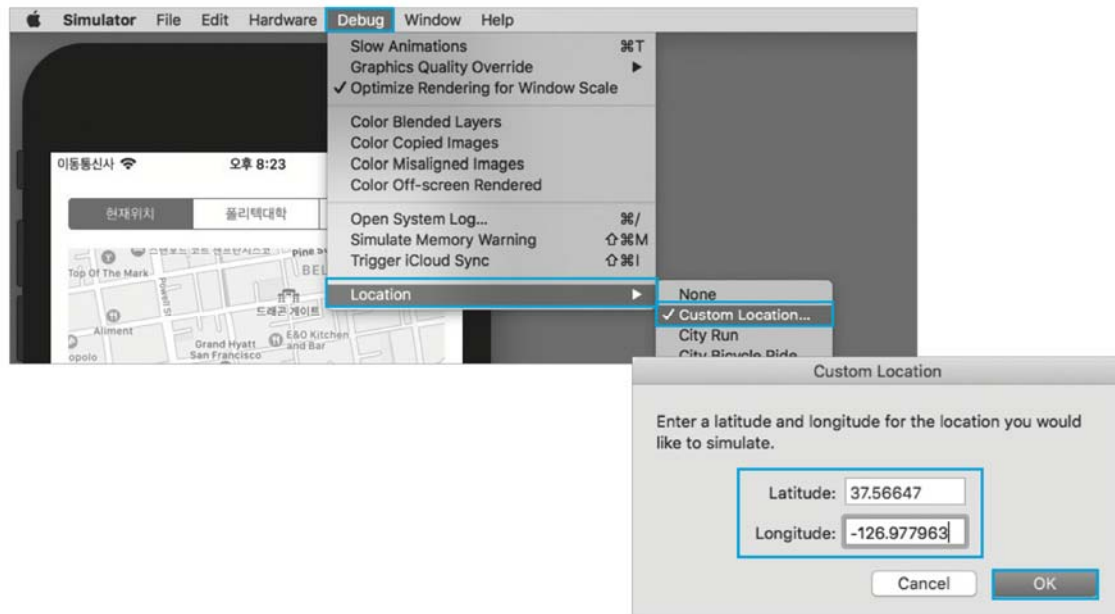
```
if flightCode[flightNumber] != nil {
    print("항공사 코드 \(flightNumber)는 \(flightCode[flightNumber]!)입니다.")
} else {
    print("없는 항공사 코드입니다.")
}
```


08-9 시뮬레이터에서 현재 위치 바꾸기

현재 위치 정보를 텍스트로 추출하여 화면에 보여 주려 했지만 시뮬레이터에서 자동으로 현재위치를 지원하지 않아 확인할 수 없었다. 시뮬레이터에서는 특정 위치를 현재 위치로 지정하여 위치정보가 정확하게 표시되는지 확인해 보자.

1. '현재 위치 정보 보여주기' 결과 보기

시뮬레이터의 메뉴에서 [Debug -> Location -> Custom Location...]을 선택한 후 위도와 경도 값을 수정한다. 여기서는 서울 시청의 위도와 경도를 사용한다.

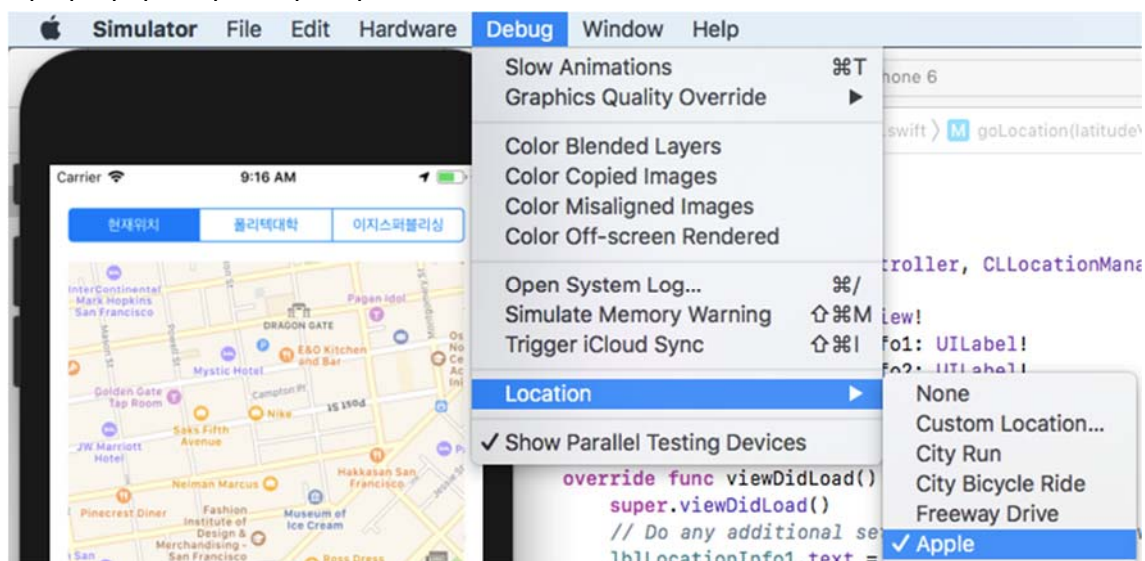


2. 앱을 다시 실행하면 지도에 서울 시청이 표시되고, 위치 정보에는 '대한민국 중구 태평로 1가'가 표시된다.



3. 이번엔 위치를 미국의 애플 사로 수정해 보겠다. 메뉴에서 [Debug -> Location -> Apple]을 선택한다.

4. 그러면 현재 위치를 알려주는 동그란 원이 사라진 것을 확인 할 수 있다. 앱을 다시 실행하면 지도에 미국 애플 사가 표시되고, 위치 정보에 '미 합중국 쿠퍼티노'가 표시된다.



4. 앱을 다시 실행하면 지도에 미국 애플 사가 표시되고, 위치정보에 '미 합중국 쿠퍼티노'가 표시된다.

특정 위치의 위도와 경도 값 찾기

<http://www.iegate.net/maps/rgeogoogle.php> 에서는 위도와 경도를 쉽게 찾을 수 있으므로 이 사이트에 접속한다. 유명한 곳이 경우에는 이름만으로도 위도, 경도 및 주소가 검색이 되지만 그렇지 않은 경우에는 주소를 입력해 위도와 경도를 찾을 수 있다.



08-10 위도와 경도로 원하는 핀 설치하기

이제는 특정 위도와 경도에 핀을 설치하고, 그 핀을 클릭하면 특정문자를 나타내도록 추가로 코드를 작성해 보자.

1. 핀 설치를 위한 초기화하기

원하는 곳에 핀을 설치하기 위해 `setAnnotation` 함수를 만든다. 입력 파라미터는 위도, 경도, 범위, 타이틀, 서브 타이틀로 하여 이 함수를 `locationManager` 함수 바로 위에 추가한다. `setAnnotation` 함수에 다음 소스를 입력한다.

```
45     func setAnnotation(latitudeValue: CLLocationDegrees, longitudeValue :  
        CLLocationDegrees, delta span :Double, title strTitle: String, subtitle  
        strSubtitle:String) {  
46         let annotation = MKPointAnnotation()  
47         annotation.coordinate = goLocation(latitudeValue: latitudeValue,  
        longitudeValue: longitudeValue, delta: span)  
48     }  
49 }
```

46 행은 핀을 설치하기 위해 `MKPointAnnotation` 함수를 호출하여 리턴 값을 `annotation` 으로 받는다.

47 행 `annotation` 의 `coordinate` 값을 `goLocation` 함수로 부터 `CLLocationCoordination2D` 형태로 받아야 하는데, 이를 위해서는 `goLocation` 함수를 수정해야 한다.

2. 앞에서 입력했던 `goLocation` 함수를 찾아 `goLocation` 의 리터 타입을 `CLLocation Coordination2D` 로 하고 코드에 직접 `return pLocation` 을 추가한다.

```
func goLocation(latitudeValue: CLLocationDegrees,  
    longitudeValue : CLLocationDegrees, delta  
    span :Double) -> CLLocationCoordinate2D {  
    let pLocation = CLLocationCoordinate2DMake(latitudeValue, longitudeValue)  
    let spanValue = MKCoordinateSpanMake(span, span)  
    let pRegion = MKCoordinateRegionMake(pLocation, spanValue)  
    myMap.setRegion(pRegion, animated: true)  
    return pLocation  
}
```

3. `goLocation` 함수를 반환값을 가지는 함수로 수정하였으므로 `locationManager` 함수 내의 `goLocation` 함수 부분을 추가한다.

핀의 타이틀과 서브 타이틀을 입력 파라미터를 이용하여 세팅하고 맵 뷰에 변수 `annotation` 값을 추가한다.

```

func setAnnotation(latitudeValue: CLLocationDegrees,
    longitudeVale : CLLocationDegrees, delta span :Double, title strTitle:
String, subtitle strSubtitle:String) {
    let annotation = MKPointAnnotation()
    annotation.coordinate = goLocation(latitude: latitudeValue,
        longitude: longitudeVale, delta: span)
    annotation.title = strTitle
    annotation.subtitle = strSubtitle
    myMap.addAnnotation(annotation)
}

```

4. 이제는 세그먼트 컨트롤러의 액션을 수정한다, '현재 위치', '폴리텍대학', '이지스퍼스팅' 이렇게 3 개를 선택할 수 있는데, 이것들이 작동하도록 코딩한다.

Sender.selectedSegmentIndex 값은 현재 위치를 기점으로 하기 때문에 '현재 위치'를 선택하면 0 이되고, '폴리텍대학'은 1, '이지스퍼블리싱'은 2 가 된다.

따라서 if 문을 이용하여 작성한다.

5. 폴리텍대학의 위치에 핀 설치하기

폴리텍 대학에 핀을 설치하기 위해 setAnnotation 함수를 호출한다, 위도, 경도, 범위, 타이틀, 서브 타이틀을 입력한다. 그리고 레이블 값도 함께 추가한다.

```

77     @IBAction func sgChangeLocation(_ sender: UISegmentedControl) {
78         if sender.selectedSegmentIndex == 0 {
79             } else if sender.selectedSegmentIndex == 1 {
80                 setAnnotation(latitudeValue: 37.751853, longitudeValue:
128.87605740000004, delta: 1, title: "한국폴리텍대학 강릉캠퍼스",
subtitle: "강원도 강릉시 남산초교길 121")
81                 self.lblLocationInfo1.text = "보고 계신 위치"
82                 self.lblLocationInfo2.text = "한국폴리텍대학 강릉캠퍼스"
83             } else if sender.selectedSegmentIndex == 2 {
84                 }
85             }
86         }

```

결과를 확인 해보면 지도의 폴리텍대학이 위치한 곳에 핀이 설치되고 레이블도 바뀐 것을 확인할 수 있다. 핀을 클릭하면 타이틀과 서브 타이틀이 나타난다.

6. 앞에서 배운 것을 응용하여 이지스퍼블리싱에도 핀을 표시하겠다. 다시 한번 setAnnotation 함수를 호출한다, 위도, 경도, 범위, 타이틀, 서브 타이틀을 입력한다 그리고 레이블 값도 함께 추가한다.


```

○ @IBAction func sgChangeLocation(_ sender: UISegmentedControl) {
77     if sender.selectedSegmentIndex == 0 {
78
79     } else if sender.selectedSegmentIndex == 1 {
80         setAnnotation(latitudeValue: 37.751853, longitudeValue:
            128.87605740000004, delta: 1, title: "한국폴리텍대학 강릉캠퍼스",
            subtitle: "강원도 강릉시 남산초교길 121")
81         self.lblLocationInfo1.text = "보고 계신 위치"
82         self.lblLocationInfo2.text = "한국폴리텍대학 강릉캠퍼스"
83     } else if sender.selectedSegmentIndex == 2 {
84         setAnnotation(latitudeValue: 37.5307871, longitudeValue: 126.8981,
            delta: 0.1, title: "이지스퍼블리싱", subtitle: "서울시 영등포구 당산로 41
            길 11")
85         self.lblLocationInfo1.text = "보고 계신 위치"
86         self.lblLocationInfo2.text = "이지스퍼블리싱 출판사 "
87     }
88 }

```

7. 마지막으로 현재 위치가 표시되도록 레이블 값을 공백으로 초기화하고, LocationManager.startUpdatingLocation 함수를 호출한다. 레이블 값을 초기화해야 기존에 작성되어 있던 텍스트를 삭제하고 현재 위치를 표시할 수 있다.

```

○ @IBAction func sgChangeLocation(_ sender: UISegmentedControl) {
77     if sender.selectedSegmentIndex == 0 {
78         self.lblLocationInfo1.text = ""
79         self.lblLocationInfo2.text = ""
80         locationManager.startUpdatingLocation()
81     } else if sender.selectedSegmentIndex == 1 {
82         setAnnotation(latitudeValue: 37.751853, longitudeValue:
            128.87605740000004, delta: 1, title: "한국폴리텍대학 강릉캠퍼스",
            subtitle: "강원도 강릉시 남산초교길 121")
83         self.lblLocationInfo1.text = "보고 계신 위치"
84         self.lblLocationInfo2.text = "한국폴리텍대학 강릉캠퍼스"
85     } else if sender.selectedSegmentIndex == 2 {
86         setAnnotation(latitudeValue: 37.5307871, longitudeValue: 126.8981,
            delta: 0.1, title: "이지스퍼블리싱", subtitle: "서울시 영등포구 당산로 41
            길 11")
87         self.lblLocationInfo1.text = "보고 계신 위치"
88         self.lblLocationInfo2.text = "이지스퍼블리싱 출판사 "
89     }

```

iOS 시뮬레이터의 [실행]클릭하여 결과를 확인한다, 시뮬레이터 메뉴에서 [Debug -> Location -> Custom Location...]을 선택하여 위치를 수정한다.

