Swift Study 04

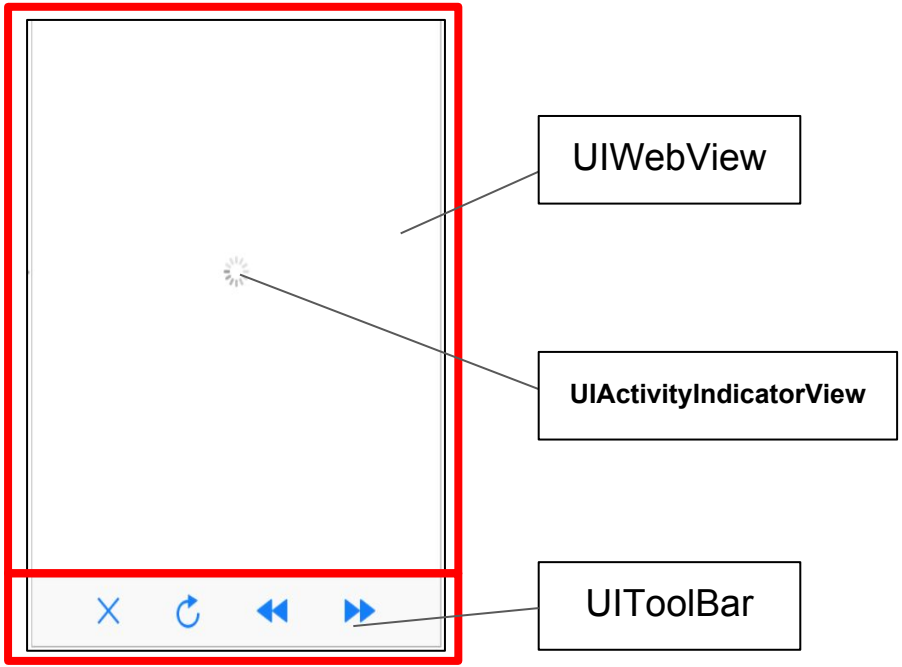
2016. 11.19

Swift UIWebView & Framework

- UIWebView
- Bundle 객체
- Mapkit / CoreLocation

UIWebView

- **html web브라우저를 지원하는 뷰**
- **UIView 하위 클래스**



UIWebView

Loading Content

```
func load(Data, mimeType: String, textEncodingName: String, baseURL: URL)
```
Sets the main page contents, MIME type, content encoding, and base URL.

```
func loadHTMLString(String, baseURL: URL?)
```
Sets the main page content and base URL.

```
func loadRequest(URLRequest)
```
Connects to a given URL by initiating an asynchronous client request.

```
var request: URLRequest?
```
The URL request identifying the location of the content to load.

```
var isLoading: Bool
```
A Boolean value indicating whether the receiver is done loading content.

```
func stopLoading()
```
Stops the loading of any web content managed by the receiver.

```
func reload()
```
Reloads the current page.

UIWebView

Moving Back and Forward

```
var canGoBack: Bool
```

A Boolean value indicating whether the receiver can move backward.

```
var canGoForward: Bool
```

A Boolean value indicating whether the receiver can move forward.

```
func goBack()
```

Loads the previous location in the back-forward list.

```
func goForward()
```

Loads the next location in the back-forward list.

UIWebView

Setting Web Content Properties

`var allowsLinkPreview: Bool`

A Boolean value that determines whether pressing on a link displays a preview of the destination for the link.

`var scalesPageToFit: Bool`

A Boolean value determining whether the webpage scales to fit the view and the user can change the scale.

`var scrollView: UIScrollView`

The scroll view associated with the web view.

`var suppressesIncrementalRendering: Bool`

A Boolean value indicating whether the web view suppresses content rendering until it is fully loaded into memory.

`var keyboardDisplayRequiresUserAction: Bool`

A Boolean value indicating whether web content can programmatically display the keyboard.

`var dataDetectorTypes: UIDataDetectorTypes`

The types of data converted to clickable URLs in the web view's content.

UIWebViewDelegate

- UIWebView에 대한 delegate protocol

Loading Content

```
func webView(UIWebView, shouldStartLoadWith: URLRequest, navigationTyp
e: UIWebViewNavigationType)
```

Sent before a web view begins loading a frame.

```
func webViewDidStartLoad(UIWebView)
```

Sent after a web view starts loading a frame.

```
func webViewDidFinishLoad(UIWebView)
```

Sent after a web view finishes loading a frame.

```
func webView(UIWebView, didFailLoadWithError: Error)
```

Sent if a web view failed to load a frame.

UIActivityIndicatorView

- 화면로딩 및 지연에 대한 애니메이션 이미지를 출력하는 **View**
- 별도의 스타일을 지정할 수 있으며, **animation**의 여부와 정지시 숨길지의 여부 설정가능

Initializing an Activity Indicator

```
init(activityIndicatorStyle: UIActivityIndicatorViewStyle)
```
Initializes and returns an activity-indicator object.

Managing an Activity Indicator

```
func startAnimating()
```
Starts the animation of the progress indicator.

```
func stopAnimating()
```
Stops the animation of the progress indicator.

```
var hidesWhenStopped: Bool
```
A Boolean value that controls whether the receiver is hidden when the animation is stopped.

Bundle 객체

- object-c / swift의 foundation framework 종류 중 하나
- 파일 자원 등이나 특정 번들 디렉토리 코드소스 접근 등을 활용하기 위한 용도
- Bundle의 main경우 현재 프로젝트 디렉토리에 대한 접근에 대한 bundle
(현재 실행중인 코드가 포함 된 번들 디렉토리를 나타냄)

```
// Get the app's main bundle
let mainBundle = Bundle.main

// Get the bundle containing the specified private class.
let myBundle = Bundle.init(for: NSClassFromString("MyPrivateClass")!)


// 특정파일 (html)의 경로를 가져오는 경우
let filePath = htmlBundle.path(forResource: "index", ofType: "html")
```

Bundle 객체

- 번들 객체는 디스크 리소스를 검색 할 때 특정 검색 패턴에 따름
- 글로벌 자원(즉, 언어 별 **.lproj** 디렉토리에없는 리소스)가 먼저 반환 이어 지역 고유의 언어 별 리소스를 반환.
- 이 검색 패턴은 번들이 다음 순서로 리소스를 검색하는 것을 의미합니다.

1.글로벌 (비 현지화) 자원

2.지역 고유의 지역화 된 리소스 (사용자의 지역 설정에 따라)

3.언어 별 지역화 된 리소스 (사용자의 언어 설정에 따라)

4.개발 언어 자원 (번들 **Info.plist** 파일 **CFBundleDevelopmentRegion** 키로 지정)

Bundle 객체 - 생성 및 초기화

Creating and Initializing a Bundle

`init(for: AnyClass)`

Returns the `NSBundle` object with which the specified class is associated.

`init?(identifier: String)`

Returns the `NSBundle` instance that has the specified bundle identifier.

`init?(url: URL)`

Returns an `NSBundle` object initialized to correspond to the specified file URL.

`init?(path: String)`

Returns an `NSBundle` object initialized to correspond to the specified directory.

Bundle 객체 - nib파일

Loading Nib Files

```
func loadNibNamed(String, owner: Any?, options: [AnyHashable : Any]? = nil)
```

Unarchives the contents of a nib file located in the receiver's bundle.

```
func loadNibNamed(String, owner: Any?, topLevelObjects: AutoreleasingUnsafeMutablePointer<NSArray>?)
```

Loads a nib from the bundle with the specified file name and owner.

Bundle 객체 - file 리소스 검색

Finding Resource Files

```
func url(forResource: String?, withExtension: String?, subdirectory: S
tring?)
```

Returns the file URL for the resource file identified by the specified name and extension and residing in a given bundle directory.

```
func url(forResource: String?, withExtension: String?)
```

Returns the file URL for the resource identified by the specified name and file extension.

```
func urls(forResourcesWithExtension: String?, subdirectory: String?)
```

Returns an array of file URLs for all resources identified by the specified file extension and located in the specified bundle subdirectory.

```
func url(forResource: String?, withExtension: String?, subdirectory: S
tring?, localization: String?)
```

Returns the file URL for the resource identified by the specified name and file extension, located in the specified bundle subdirectory, and limited to global resources and those associated with the specified localization.

Bundle 객체 - file 리소스 검색

```
func path(forResource: String?, ofType: String?)
```
Returns the full pathname for the resource identified by the specified name and file extension.

```
func path(forResource: String?, ofType: String?, inDirectory: String?)
```
Returns the full pathname for the resource identified by the specified name and file extension and located in the specified bundle subdirectory.

```
func path(forResource: String?, ofType: String?, inDirectory: String?,
forLocalization: String?)
```
Returns the full pathname for the resource identified by the specified name and file extension, located in the specified bundle subdirectory, and limited to global resources and those associated with the specified localization.

```
func paths(forResourcesOfType: String?, inDirectory: String?)
```
Returns an array containing the pathnames for all bundle resources having the specified filename extension and residing in the resource subdirectory.

```
func paths(forResourcesOfType: String?, inDirectory: String?, forLocal
ization: String?)
```
Returns an array containing the file for all bundle resources having the specified filename extension, residing in the specified resource subdirectory, and limited to global resources and those associated with the specified localization.

Bundle 객체 - 이미지 리소스 검색

Finding Image Resources

`func urlForImageResource(String)`

Returns the location of the specified image resource as an NSURL.

`func pathForImageResource(String)`

Returns the location of the specified image resource file.

`func image(forResource: String)`

Returns an `NSImage` instance associated with the specified name, which can be backed by multiple files representing different resolution versions of the image.

Bundle 객체 - 기타정보들

Finding Sound Resources

```
func path(forSoundResource: String)
```
Returns the location of the specified sound resource file.

Fetching Localized Strings

```
func localizedString(forKey: String, value: String?, table: String?)
```
Returns a localized version of the string designated by the specified key and residing in the specified table.

Fetching Context Help Resources

```
func contextHelp(forKey: String)
```
Returns the context-sensitive help for the specified key from the bundle's help file.

Getting the Standard Bundle Directories

```
var resourceURL: URL?
```
The file URL of the bundle's subdirectory containing resource files.

```
var executableURL: URL?
```
The file URL of the receiver's executable file.

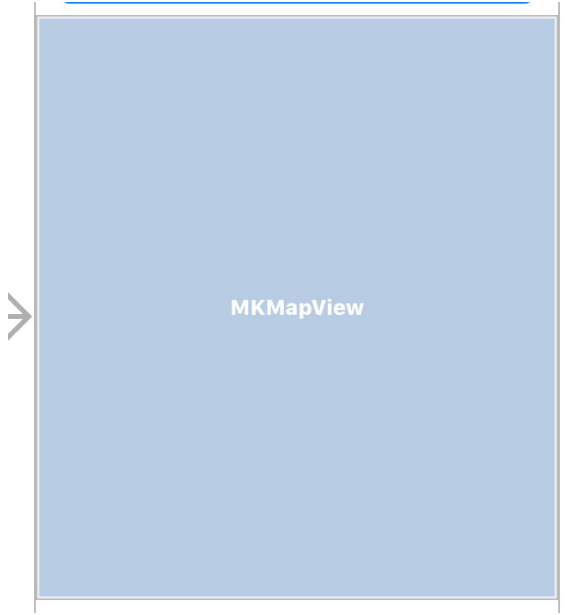
```
var privateFrameworksURL: URL?
```
The file URL of the bundle's subdirectory containing private frameworks.

Mapkit

- 지도정보를 위한 **framework**
- 다양한 클래스 중 **MKMapView**를 이용하여 **MapView** 사용
- 기본클래스에서는 **UIKit**만 **import**되어있기 때문에 별로도 **import MapKit** 선언 필요

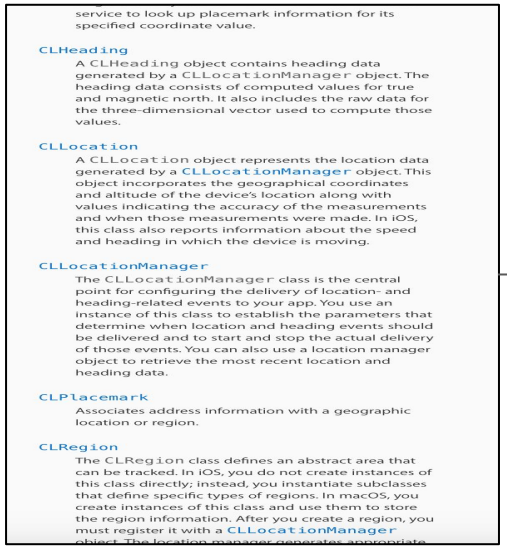
```
import Foundation
import MapKit.MKAnnotation
import MapKit.MKAnnotationView
import MapKit.MKCircle
import MapKit.MKCircleRenderer
import MapKit.MKCircleView
import MapKit.MKDirections
import MapKit.MKDirectionsRequest
import MapKit.MKDirectionsResponse
import MapKit.MKDirectionsTypes
import MapKit.MKDistanceFormatter
import MapKit.MKFoundation
import MapKit.MKGeodesicPolyline
import MapKit.MKGeometry
import MapKit.MKLocalSearch
import MapKit.MKLocalSearchCompleter
import MapKit.MKLocalSearchRequest
import MapKit.MKLocalSearchResponse
import MapKit.MKMapCamera
import MapKit.MKMapItem
import MapKit.MKMapSnapshot
import MapKit.MKMapSnapshotOptions
import MapKit.MKMapSnapshotter
import MapKit.MKMapView
import MapKit.MKMultiPoint
import MapKit.MKOverlay
import MapKit.MKOverlayPathRenderer
import MapKit.MKOverlayPathView
import MapKit.MKOverlayRenderer
import MapKit.MKOverlayView
import MapKit.MKPinAnnotationView
import MapKit.MKPlacemark
import MapKit.MKPointAnnotation
import MapKit.MKPolygon
import MapKit.MKPolygonRenderer
import MapKit.MKPolygonView
import MapKit.MKPolyline
import MapKit.MKPolylineRenderer
import MapKit.MKPolylineView
import MapKit.MKReverseGeocoder
import MapKit.MKTileOverlay
import MapKit.MKTileOverlayRenderer
import MapKit.MKTypes
import MapKit.MKUserLocation
import MapKit.MKUserTrackingBarButtonItem
import MapKit.NSUserActivity_MKMapItem
import MapKit
```

MKMapView



CoreLocation

- 장치의 현재 위도와 경도를 결정하여 위치 관련 이벤트의 전달을 설정 및 예약할 수 있는 프레임워크
- 기본적으로 **MapKit**에 내장 **import** 되어 있음
- 예제에서는 위치에 대한 측위정보를 **CoreLocation**의 하위**class** 중 **CLLocationManager**를 사용


service to look up placemark information for its specified coordinate value.

`CLHeading`
A `CLHeading` object contains heading data generated by a `CLLocationManager` object. The heading data consists of computed values for true and magnetic north. It also includes the raw data for the three-dimensional vector used to compute those values.

`CLLocation`
A `CLLocation` object represents the location data generated by a `CLLocationManager` object. This object incorporates the geographical coordinates and altitude of the device's location along with values indicating the accuracy of the measurements and when those measurements were made. In iOS, this class also reports information about the speed and heading in which the device is moving.

`CLLocationManager`
The `CLLocationManager` class is the central point for configuring the delivery of location- and heading-related events to your app. You use an instance of this class to establish the parameters that determine when location and heading events should be delivered and to start and stop the actual delivery of those events. You can also use a location manager object to retrieve the most recent location and heading data.

`CLPlacemark`
Associates address information with a geographic location or region.

`CLRegion`
The `CLRegion` class defines an abstract area that can be tracked. In iOS, you do not create instances of this class directly; instead, you instantiate subclasses that define specific types of regions. In macOS, you create instances of this class and use them to store the region information. After you create a region, you must register it with a `CLLocationManager` object. The location manager generates appropriate

CLLocationManager

- 위치 이벤트와 제목 이벤트를 전달하는시기를 결정하는 클래스
- 실제 위치정보에 대한 시작/중지 설정가능, 또한 최신의 위치와 주소정보를 검색 할 수 있음.

Initiating Standard Location Updates

```
func startUpdatingLocation()
```
Starts the generation of updates that report the user's current location.

```
func stopUpdatingLocation()
```
Stops the generation of location updates.

```
func requestLocation()
```
Request the one-time delivery of the user's current location.

```
var pausesLocationUpdatesAutomatically: Bool
```
A Boolean value indicating whether the location manager object may pause location updates.

```
var allowsBackgroundLocationUpdates: Bool
```
A Boolean value indicating whether the app wants to receive location updates when suspended.

```
var distanceFilter: CLLocationDistance
```
The minimum distance (measured in meters) a device must move horizontally before an update event is generated.

```
var desiredAccuracy: CLLocationAccuracy
```
The accuracy of the location data.

```
var activityType: CLActivityType
```
The type of user activity associated with the location updates.

CLLocationManagerDelegate

- CLLocationManager의 이벤트별 delegate

```
func locationManager(_ manager: CLLocationManager, didUpdateLocations locations: [CLLocation])  //위치정보 변경에 대한 이벤트
func locationManager(_ manager: CLLocationManager, didUpdateHeading newHeading: CLHeading)
func locationManagerShouldDisplayHeadingCalibration(_ manager: CLLocationManager) -> Bool
func locationManager(_ manager: CLLocationManager, didDetermineState state: CLRegionState, for region: CLRegion)
func locationManager(_ manager: CLLocationManager, didRangeBeacons beacons: [CLBeacon], in region: CLBeaconRegion)
func locationManager(_ manager: CLLocationManager, rangingBeaconsDidFailFor region: CLBeaconRegion, withError error: Error)
func locationManager(_ manager: CLLocationManager, didEnterRegion region: CLRegion)
func locationManager(_ manager: CLLocationManager, didExitRegion region: CLRegion)
func locationManager(_ manager: CLLocationManager, didFailWithError error: Error)
func locationManager(_ manager: CLLocationManager, monitoringDidFailFor region: CLRegion?, withError error: Error)
func locationManager(_ manager: CLLocationManager, didChangeAuthorization status: CLAuthorizationStatus)
func locationManager(_ manager: CLLocationManager, didStartMonitoringFor region: CLRegion)
func locationManagerDidPauseLocationUpdates(_ manager: CLLocationManager)
func locationManagerDidResumeLocationUpdates(_ manager: CLLocationManager)
func locationManager(_ manager: CLLocationManager, didFinishDeferredUpdatesWithError error: Error?)
func locationManager(_ manager: CLLocationManager, didVisit visit: CLVisit)
```