# Found
# Lesson 3

## Description

Modify the Map View attributes and use the Core Location framework to display a location beacon. Configure the iOS Simulator with a custom location.


## Learning Outcomes

- Practice using the Attributes Inspector to configure view components.
- Discover how to use the Core Location framework to display a location beacon.
- Discover how to configure a device location with the iOS Simulator.

## Vocabulary

| | | |
|---|---|---|
| Attributes Inspector | location services | framework |
| Core Location | `CLLocationManager` | app delegate |
| Info.plist | latitude | longitude |

## Materials

- **Found Lesson 3** Xcode project
- Latitude and longitude of your school, such as 39.7508,-105.2238

# Opening

How can we pinpoint our current location on the map?

# Agenda

- Discuss the objective of displaying the current location of the device on the map.
- Using Interface builder, select the Map View and open the Attributes Inspector (⌥⌘4).
- Change the Type to **Hybrid** and ensure that Shows **User Location** is checked.
- Run the app (⌘R), and observe how the map adds satellite imagery on the map.
- Observe the console warning **Trying to start MapKit location updates… must call requestWhenInUseAuthorization … first**.
- Explain how iOS apps must request user authorization to use location information with the Core Location framework.
- Import the Core Location framework above the `AppDelegate` class definition.

```
import UIKit
import CoreLocation
...
```

- Using the Xcode Documentation and API Reference (⇧⌘0), search for **Core Location Framework** and explore some of the resulting documentation.
- In the `AppDelegate` class, declare a `CLLocationManager` property with a default value.

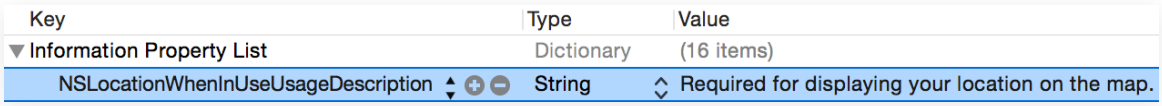
```
let locationManager = CLLocationManager()
```

- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `CLLocationManager` class.
- In the `AppDelegate` class, modify `application:didFinishLaunchingWithOptions:` to request permission to use iOS location services.

```
func application(application: UIApplication,
   didFinishLaunchingWithOptions launchOptions:[NSObject: AnyObject]?)
   -> Bool {
   locationManager.requestWhenInUseAuthorization()
   return true
}
```

- Explain how the `application:didFinishLaunchingWithOptions:` method will prompt the user for permission to use location services, but will require additional app configuration.

- Using the Project Navigator (⌘1), open **Info.plist** and add the Key `NSLocationWhenInUseUsageDescription` and `Value Required for displaying your location on the map`.




- Run the app (⌘R), and tap the Allow button. Observe the position of the location beacon, scrolling and zooming the map by ⌥-clicking with the mouse if necessary.
- Discuss that the iOS Simulator chooses Cupertino, CA as the default location.
- Using the Simulator menu item *Debug* > *Location* > *Custom Location…*, enter a latitude and longitude, and observe the change in the position of the location beacon.

# Closing

How do latitude and longitude values relate to locations on the Earth?

# Modifications and Extensions

- Create a controller outlet for the map view, investigate the `MKMapView` class reference, and, instead of using the Attributes Inspector, set the map view properties with controller code in `viewDidLoad`.
- Move the `CLLocationManager` out of the `AppDelegate` and into the `ViewController` class. Explain which approach seems better for this app.
- Delete the app from the Simulator, run it again, and wait a few moments before tapping the Allow button when prompted for allowing location services. Observe the console output, and investigate why the warning message still appears. Improve the codebase by incorporating a `CLLocationManagerDelegate`, and use appropriate delegate methods to control the app behavior to squelch the console warning.

# Resources

Configuring Object Attributes https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/ObjectAttributes.html

Core Location Framework Reference https://developer.apple.com/library/ios/documentation/CoreLocation/Reference/CoreLocation_Framework/

UIApplicationDelegate Protocol Reference https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIApplicationDelegate_Protocol/index.html

iOS Simulator User Guide https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/iOS_Simulator_Guide/