

# Gesture Recognizer Basics

Gesture recognizers are the simplest way to handle touch or press events in your views. You can attach one or more gesture recognizers to any view. Gesture recognizers encapsulate all of the logic needed to process and interpret incoming events for that view and match them to a known pattern. When a match is detected, the gesture recognizer notifies its assigned target object, which can be a view controller, the view itself, or any other object in your app. Notifications are sent using the target-action design pattern, an example of which is shown in Figure 1-1. When the [UITapGestureRecognizer](#) object detects a single-finger tap in the view, it calls an action method of the view's view controller, which you use to provide a response.

제스처 인식기 (gesture recognizer) 는 뷰에서 일어난 터치 또는 프레스 이벤트를 처리하는 가장 간단한 방법입니다. 뷰에는 하나 이상의 제스처 인식기가 첨부될 수 있습니다. 제스처 인식기는 해당 뷰에 들어오는 이벤트를 해석하고 그것을 알려진 패턴으로 매칭시키는 데 필요한 모든 로직을 캡슐화합니다. 매칭이 감지되었을 때, 제스처 인식기는 할당된 타겟 객체에 이를 통보하며, 타겟 객체는 뷰 컨트롤러, 뷰 자체 또는 다른 객체가 될 수 있습니다. 아래의 그림 1-1의 예처럼, 노티피케이션은 타겟-액션 디자인 패턴을 사용하여 전송됩니다. [UITapGestureRecognizer](#) 객체가 탭 이벤트를 감지하였을 때, 뷰가 속한 뷰 컨트롤러의 액션 메소드를 호출합니다.

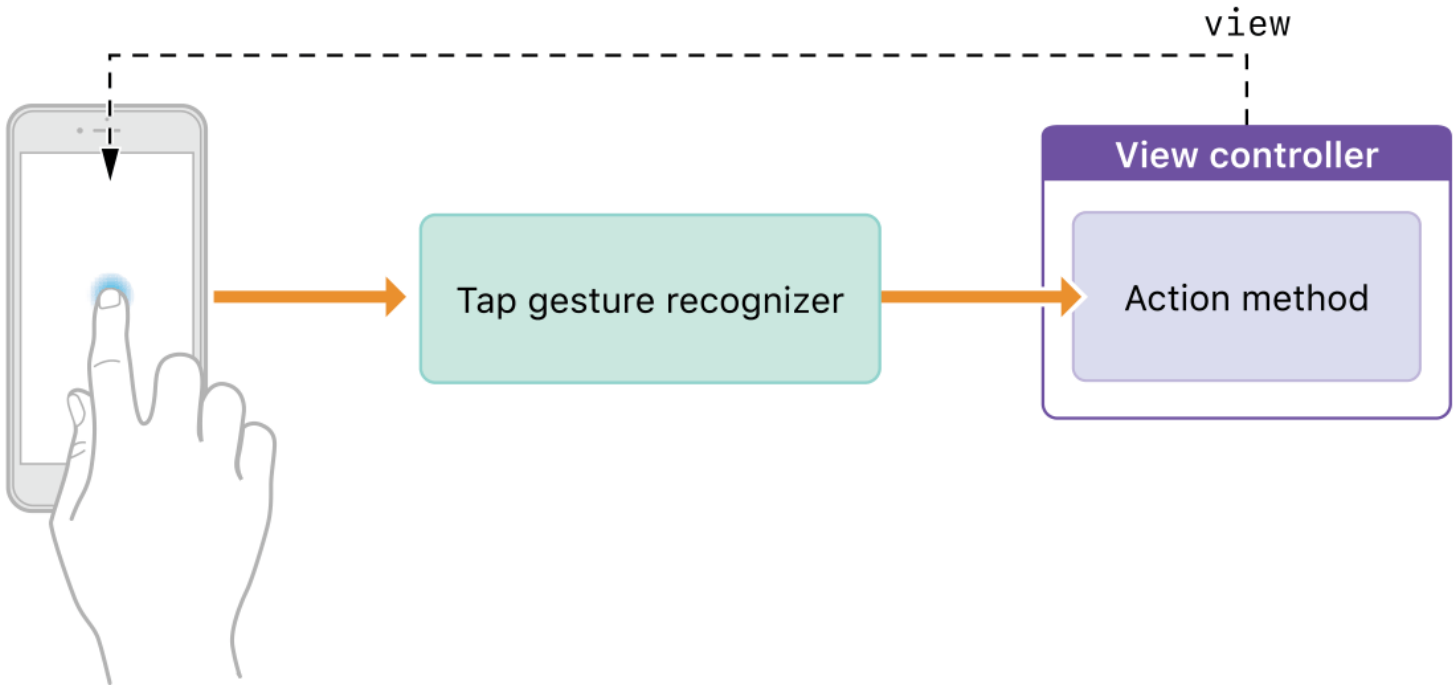


Figure 1-1A gesture recognizer notifying its target

Gesture recognizers come in two types: discrete and continuous. A *discrete gesture recognizer* calls your action method once after the gesture is recognized. A *continuous gesture recognizer* may call your action method many times, including at the beginning and end of the gesture and each time the details of a tracked event change. For example, a [UIPanGestureRecognizer](#) object calls your action method whenever the position of a touch event changes.

제스처 인식기는 discrete 그리고 continuous 두 가지 형태로 제공됩니다. discrete 제스처 인식기는 제스처가 인식되고 난 후 액션메소드를 한 번 호출합니다. continuous 제스처 인식기는 제스처의 시작과 끝 그리고 이벤트 변화에 따른 세부사항을 포함하여 여러번 호출될 수 있습니다. 예를 들어, [UIPanGestureRecognizer](#) 객체는 터치 이벤트의 위치가 변경 될 때마다 액션 메소드를 호출합니다.

Interface Builder includes objects for each of the standard UIKit gesture recognizers. It also includes a custom gesture recognizer object that you can use to represent your custom [UIGestureRecognizer](#) subclasses.

인터페이스 빌더는 표준 UIKit 제스처 인식이 각각에 대한 객체를 포함하고 있습니다. 또한 [UIGestureRecognizer](#)을 서브클래싱한 커스텀 제스처 인식기도 포함됩니다.

### To configure a gesture recognizer

제스처 인식을 구성하려면..

- In your storyboard, drag the gesture recognizer onto your view.  
스토리보드에서 제스처 인식을 뷰에 드래그 하세요.
- Implement an action method to be called when the gesture is recognized; see [Listing 1-1](#).  
제스처가 인식되었을 때 호출될 액션 메소드를 구현하세요.
- Connect your action method to the gesture recognizer.  
액션메소드를 제스처 인식기와 연결하세요.

You can create this connection in Interface Builder by right-clicking the gesture recognizer and connecting its Sent Action selector to the appropriate object in your interface. You can also configure the action method programmatically using the [addTarget:action:](#) method of the gesture recognizer. Listing 1-1 shows the generic format for the action method of a gesture recognizer. If you prefer, you can change the parameter type to match a specific gesture recognizer subclass.

이러한 연결을 인터페이스에서 제스처 인식을 우클릭한 후 적절한 객체의 액션 셀렉터와 연결하여 생성할 수 있습니다. 또한 제스처 인식기의 [addTarget:action:](#) 메소드를 이용하여 코드로 액션 메소드를 설정 할 수도 있습니다. 아래 리스트 1-1은 제스처 인식기의 액션 메소드에 대한 일반적인 포맷을 보여줍니다. 원하는 경우 제스처 인식기의 서브클래스와 매칭시키기 위해 파라미터 타입을 변경 할 수 있습니다.

### Listing 1-1 제스처 인식기 동작 메소드

```
- (IBAction)myActionMethod:(UIGestureRecognizer*)sender

@IBAction func myActionMethod(_ sender: UIGestureRecognizer)
```

## Responding to Gestures

The action method associated with a gesture recognizer provides your app's response to that gesture. For discrete gestures, your action method is similar to the action method for a button. Once the action method is called, you perform whatever task is appropriate for that gesture. For continuous gestures, your action method can respond to the recognition of the gesture, but it can also track events before the gesture is recognized. Tracking events lets you create a more interactive experience. For example, you might use the updates from a [UIPanGestureRecognizer](#) object to reposition content in your app.

제스처 인식기와 관련된 액션 메소드는 특정 제스처에 대한 앱의 반응을 제공합니다. discrete 제스처의 경우, 액션 메소드는 버튼의 액션 메소드와 비슷합니다. 액션 메소드가 호출 될 때, 해당 제스처에 대한 적절한 작업을 수행합니다. continuous 제스처의 경우, 액션 메소드는 제스처의 인식에 따라 응답할 수 있지만 제스처가 인식되기 전에 이벤트를 추적할 수 있습니다. 이벤트를 추적하는 것은 사용자에게 보다 인터랙티브한 경험을 제공할 수 있게 합니다. 예를 들어, 앱의 콘텐츠의 위치를 [UIPanGestureRecognizer](#) 객체를 통해 업데이트 할 수 있습니다.

The [state](#) property of a gesture recognizer communicates the object's current state of recognition. For continuous gestures, the gesture recognizer updates the value of this property from [UIGestureRecognizerStateBegan](#) to [UIGestureRecognizerStateChanged](#) to [UIGestureRecognizerStateEnded](#). Your action methods use this property to determine an appropriate course of action. For example, you might use the began and changed states to make temporary changes to your content and use the ended state to make those changes permanent. Always check the value of the state property of a gesture recognizer before taking actions.

제스처 인식기의 [state](#) 프로퍼티는 객체의 현재 인식 상태를 전달합니다. continuous 제스처의 경우, 제스처 인식기는 [UIGestureRecognizerStateBegan](#), [UIGestureRecognizerStateChanged](#) , [UIGestureRecognizerStateEnded](#)을 통해 [state](#) 프로퍼티의 값을 업데이트합니다. 액션 메소드는 이 프로퍼티를 이용하여 액션에 대한 적절한 기능을 결정 할 수 있습니다. 예를 들어, began과 changed 상태를 이용하여 콘텐츠의 일시적인 변화를 만들어 낼 수 있고 ended 상태를 이용하여 영구적인 변화를 만들 수 있습니다. 항상 액션이 일어나기 전에 제스처 인식기의 state 프로퍼티의 값을 확인하세요.

For examples of how to handle specific types of gestures, see the following information:

제스처의 특정 타입 처리에 대한 예는, 아래를 참고하세요.

[Handling Tap and Long-Press Gestures](#)

[Handling Pan and Swipe Gestures](#)

[Handling Pinch and Rotation Gestures](#)

For more information about gesture recognizer states and how they affect your code, see [Implementing a Custom Gesture Recognizer](#).

제스처 인식기의 상태와 그것에 코드에 어떠한 영향을 미치는 지에 대한 자세한 정보는 [Implementing a Custom Gesture Recognizer](#).을 참고하세요.