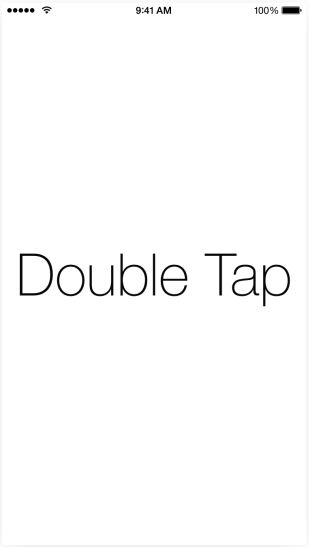
# Gesturizer
## Lesson 2

## Description

Add and configure a second gesture recognizer to detect double-taps.




## Learning Outcomes

- Practice adding objects to the user interface, modifying interface object attributes, and establishing controller outlet and action connections.
- Apply a Tap Gesture Recognizer to update the interface when the screen is double-tapped.
- Practice modifying interface components from controller code.

## Vocabulary

| | | |
|---|---|---|
| gesture | `UIGestureRecognizer` | Tap Gesture Recognizer |
| Attributes Inspector | controller action | Document Outline |
| connection well | Connections Inspector | |

## Materials

- **Gesturizer Lesson 2** Xcode project

## Opening

How might you recognize both single taps and double taps?

# Agenda

- Using Interface Builder and the Object Library (⌥⌘L), and a new Tap Gesture Recognizer to the Document Outline (◧).
- Select the new recognizer, and name it **Double Tap**.
- Using the Attributes Inspector (⌥⌘4), change the *Taps* attribute to 2.
- Using the Assistant Editor (⌥⌘↩), Control-drag a connection from the Double Tap gesture recognizer to a new controller action called `doubleTap:`.

```
@IBAction func doubleTap(sender: UITapGestureRecognizer) {
   gestureName.text = "Double Tap"
   gestureName.hidden = false
}
```

- Using Interface Builder and the Document Outline (◧), Control-click the View and drag a connection from the gestureRecognizers connection well to the Double Tap gesture recognizer, adding it to the `gestureRecognizers` outlet collection.
- With the View still selected, open the Connections Inspector (⌥⌘6) and observe the Outlet Connections.
- Discuss how the View `gestureRecognizers` collection represents a collection of gesture recognizers to which the view may be connected.
- Run the app (⌘R), tap and double-tap the screen, and observe the different label text appear.

# Closing

Can you think of a way to make our gesture label gracefully appear and then disappear after each gesture?

# Modifications And Extensions

- Instead of using two independent controller methods, set the controller as each gesture recognizer's delegate, and let the controller adopt the `UIGestureRecognizerDelegate` protocol. Implement a single protocol method that decides how to change the gesture label.

# Resources

Event Handling Guide for iOS http://developer.apple.com/library/ios/documentation/EventHandling/Conceptual/EventHandlingiPhoneOS/Introduction/Introduction.html

UIGestureRecognizer Class Reference https://developer.apple.com/library/prerelease/ios/documentation/UIKit/Reference/UIGestureRecognizer_Class/index.html

UITapGestureRecognizer Class Reference http://developer.apple.com/library/ios/documentation/uikit/reference/UITapGestureRecognizer_Class/Reference/Reference.html

Adding an Object to Your Interface https://developer.apple.com/library/ios/recipes/xcode_help-IB_objects_media/Chapters/AddingObject.html

Creating an Action Connection https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingAction.html

Cocoa Core Competencies: Target-Action http://developer.apple.com/library/ios/documentation/General/Conceptual/Devpedia-CocoaApp/TargetAction.html