**Connections I**

**Now that we finished building our UI and properly laid it out using Auto Layout, we can start thinking about the implementation of the app**. First, let's make the necessary connections between our scene and our View Controller code.

**Outlets**

We are going to create two Outlets. **We want an Outlet for the question label and the answer label because we want to be able to change the text inside of these labels in our application**. For example, if the user clicks next, we want to display the text of the next question. We do this by control-click (or right-click) dragging from the storyboard to the View Controller.

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

**Now that the labels are wired up, we can change its value in our code.** The labels have a property called text that we can change. **The viewDidLoad() method gets called automatically once our view has loaded**. This is a good place to do some set-up code. Once this message is sent to our View Controller, we will change the text properties of the labels which will result in our application displaying the correct question and answer.

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

# Connections II

Setting up an Outlet is**connecting a View object in our storyboard to a property in our code**. Setting up an Action is similar but it is**connecting an event that a user can trigger by interacting with one or more View objects to a method in our code**.

## Actions

Now let's set up Actions for our buttons. **Once the user touches the button and releases the finger within the bounds of the button, we want to trigger some lines of code**. For example, when the Answer button is clicked, we might want to display our Answer label. We specify that the View object that is going to trigger this event is going to be a UIButton. The default is AnyObject but that is too vague. **We want to explicitly state that a UIButton is going to trigger this event.** Then when this code is triggered, the sender object will be passed into the block of code which would be the UIButton that alerted the View Controller about an event.

Here, we specify the Type to be UIButton only because this describes our IBAction a little better. We are saying that this action is going to get triggered by a UIButton. Once the method gets called, it passes a sender argument which is the View object that triggered this action. We can leave it at AnyObject and our code will still work but it is good practice to be more descriptive when we can.

صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

Now we can write the lines of code that we want to execute when someone clicks inside our different buttons. We will just log out that a particular button was pressed for now. We will be able to see the results of the log in our Debug Area of our Xcode.صورة تحتوي على نص

تم إنشاء الوصف تلقائياً

# UIKit

After using Labels and Buttons you may have felt overwhelmed by the many features provided by these two elements alone or the many other options for UI elements that are avaliable. The reality is that many of these elements are quite similar to one another just specialized for specific instances. As such we feel it is only proper that we give a short breakdown of some of the different elements you can make use of and their more commonly used features.

These elements are broken down into two sections Outlets and Actions. **That is not to say that all the these elements can't be used for both IBOutlets and IBActions**. Instead the Outlets elements are more commonly used for displaying some form data which would be accessed through an IBOutlet. The Actions elements on the other hand are most commonly used for event listeners such as clicks or swipes

## Outlets

### View

### Label

### Textfield

### Image

## Actions

### Button

### Switch

### Stepper

### Segmented

### Slider

### Picker