

# **Introduction to iOS Development**

## **Tutorial Notes - Building a Temperature Calculator**



For information about training courses available from Shiny Development or for information about bespoke iOS training courses, email [contact@shinydevelopment.com](mailto:contact@shinydevelopment.com) or visit <http://shinydevelopment.com/training>.

All training materials are ©2009-2011 Shiny Development Ltd. and can not be reproduced in part or whole without the express written permission of Shiny Development Ltd.

Document version: 1.1

Shiny Development Ltd. is registered in England, Registration Number: 05805802.

Shiny Development Ltd.  
Unit CLP107  
24 Liverpool Road  
Chester, CH2 3NE.

# Building a Simple Temperature Calculator for iOS

This tutorial is a walkthrough of all of the steps needed to build a simple iOS application to run on an iPhone or iPod Touch device. The application will allow a user to enter a temperature in degrees fahrenheit, click a “Convert” button and display the temperature converted to degrees celsius.

This tutorial starts from a partially written template for this application which should have been provided along with this document.

1. Locate the TemperatureCalculator.xcodeproj file in the Template directory and double click it to load the template project with Xcode.

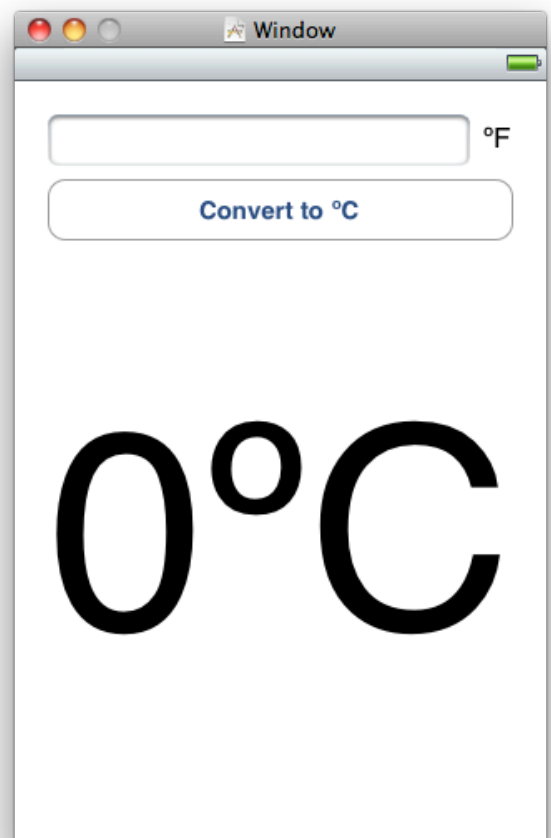
## Laying out the Interface

First thing to do is to lay out the user interface for the app. Locate the MainWindow.xib file in the Resources folder of the Groups & Files side panel on the left hand side of the Xcode window.

2. Double click the MainWindow.xib file to open it with Interface Builder.
3. Open the Library palette by selecting Library from the Tools menu (it may already be open).
4. Open the application window by double clicking the Window item in the MainWindow.xib file. The window will be blank and white for now
5. Drag a Round Rect Button, a Text Field and two Label controls from the Library palette to the application window.
6. Open the properties inspector by selecting Attributes Inspector from the Tools menu.
7. Set properties on the label, button and text field to make them look something like the screenshot to the right. Take special care to set the following properties:

- 7.1. Set the keyboard type to Number Pad in the Text Input Traits for the text field.

- 7.2. Change the font size of the text field control to be larger than the default. The default font size is too small to read on a mobile device. 18pt is a good size.



## Connecting the Interface to the Code

The next step is to hook up the interface to the code. The template project has some actions and outlets already set up ready to use so all that needs to be done is to use Interface Builder to connect them.

8. Right click on the Temperature Calculator App Delegate in the MainWindow.xib file. A floating window should appear with sections titled Outlets and Received Actions.

9. To connect the outlets:

- 9.1. Drag from the small white circle on the right hand side next to celsiusLabel and drop it on the Label control that was added to the window.
- 9.2. Drag from the circle next to the fahrenheitTextField to the Text Field control in the window.

### Remember

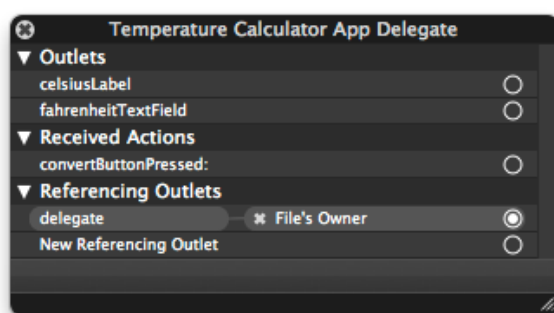
★ **Outlets** are used to connect objects in the interface to properties in code.

★ **Actions** are used to initiate events in code from actions within the user interface.

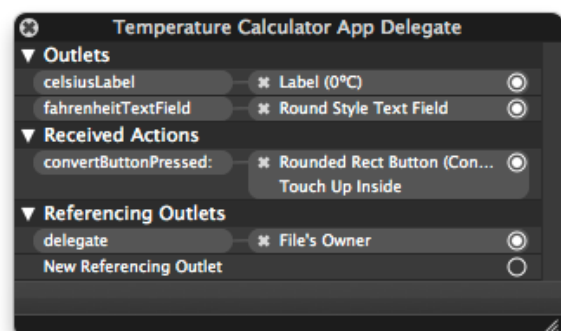
10. To connect the action:

- 10.1. Drag from the circle next to convertButtonPressed: to the button in the window and a floating menu will pop up, select Touch Up Inside from this menu.

11. After the outlets and actions are connected the floating outlets and actions window should look similar to the picture on the right hand side below:



Before



After

12. Save the MainWindow.xib file by selecting Save from the File menu.

## Writing the Code

Finally, time to write some code! As a result of the action that was hooked up in the previous section code needs to be added to read the value entered by the user in the text field, convert it from degrees fahrenheit to degrees celsius and then place the calculated value into the label as a string.

13. Back in Xcode, find the Classes folder of the Groups & Files side panel and open the TemperatureCalculatorAppDelegate.m file and locate the convertButtonPressed: method.

14. The first step is to make sure that the keyboard used to enter the the temperature is not still visible. Type the following code which will hide the keyboard (if it is visible):

```
[self.fahrenheitTextField resignFirstResponder];
```

15. Next the text from the text field in the interface needs to be placed into a string variable:

```
NSString *fahrenheitText = self.fahrenheitTextField.text;
```

16. Then the string needs to be converted into a number so that calculations can be performed on it:

```
CGFloat fahrenheit = [fahrenheitText floatValue];
```

17. Using the method which was provided by the template, next step is to convert the value to degrees celsius:

```
CGFloat celsius = [self celsiusFromFahrenheit:fahrenheit];
```

18. Again, using another method provided by the template the next step is to convert the new value back into a string so it can be put back in the user interface:

```
NSString *celsiusText = [self formattedCelsiusString:celsius];
```

19. Finally, the new string is set to be the text value of the label in the user interface:

```
self.celsiusLabel.text = celsiusText;
```

20. Run the application by clicking the Build and Run toolbar button!