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 or visit http://shinydevelopment.com/training.

Document version: 1.2

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

Shiny Development Ltd.
Daresbury Innovation Centre
Keckwick Lane, Daresbury
Cheshire, WA4 4FS
United Kingdom

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.

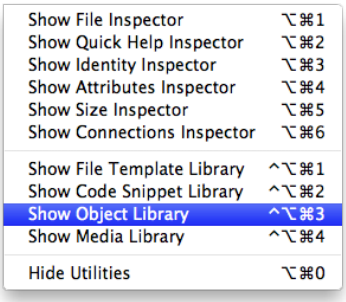
Laying out the Interface

The first thing to do is to lay out the user interface for the app.

2. Locate the MainWindow.xib file in the TemperatureCalculator folder of the side panel on the left hand side of the Xcode window.

3. Single click the MainWindow.xib file to open it in design mode.

4. Show the library of available controls by selecting the "Utilities" option from the "View" menu and then choosing "Show Object Library". It will open on the bottom right hand side of the window, although it may already be visible if you have used Xcode before.


Show File Inspector ⌥⌘1
Show Quick Help Inspector ⌥⌘2
Show Identity Inspector ⌥⌘3
Show Attributes Inspector ⌥⌘4
Show Size Inspector ⌥⌘5
Show Connections Inspector ⌥⌘6
Show File Template Library ^⌥⌘1
Show Code Snippet Library ^⌥⌘2
Show Object Library ^⌥⌘3
Show Media Library ^⌥⌘4
Hide Utilities ⌥⌘0

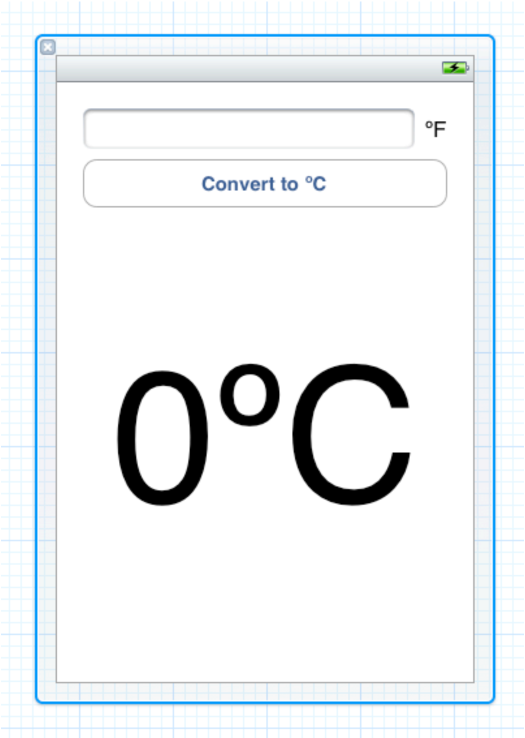
5. Drag a Round Rect Button, a Text Field and two Label controls from the object library to the application window in the centre of the design area.

6. Open the attributes inspector by selecting "Show Attributes Inspector" from the "View" menu under "Utilities".

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:

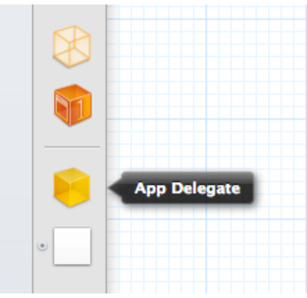
   7.1. Set the keyboard type to Number Pad in the Text Input Traits for the text field.

   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 "App Delegate" on the left hand side of the MainWindow.xib design area. The delegate is represented by a yellow/orange solid cube icon (see right). A floating black window should appear with the title "App Delegate".

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.

10. To connect the action:

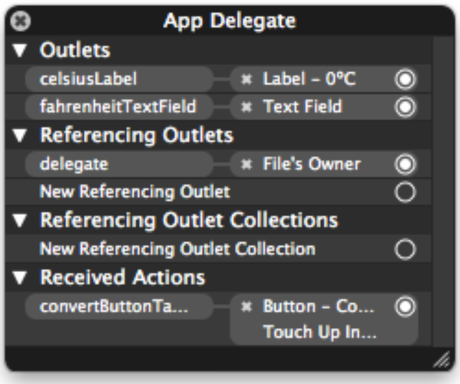
    10.1. Drag from the circle next to "convertButtonPressed:" to the button in the window and a floating menu will pop up, select "Touch Up Inside" from this menu.

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:

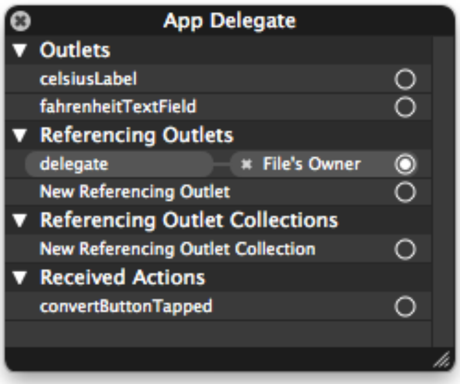
Remember

★ **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.



Before



After

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.

12. Locate the AppDelegate.m file on the left hand side panel and position the text cursor just after the green text marked with "INSERT YOUR CODE HERE".

13. The first task of this code 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:

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

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

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

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

```
double fahrenheit = [fahrenheitText doubleValue];
```

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

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

17. 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];
```

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

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

19. Run the application by clicking the Run toolbar button!