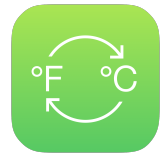


# UnitConverter

## Lesson 9

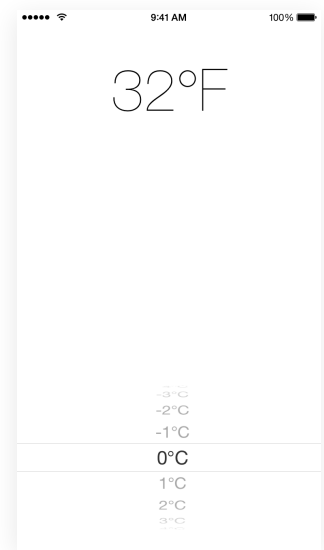


### Description

Address a usability issue when starting the app, by implementing a default starting temperature.

### Learning Outcomes

- Discuss the importance of user experience and software usability, and identify usability problems in an app.
- Recognize the benefit of calculating values from existing dependencies as opposed to using "hard-coded" values.
- Practice creating outlet connections with controller properties.
- Recognize the benefit of re-using existing functions as opposed to duplicating code.
- Solve usability issues in an app by implementing behavior with code.



### Vocabulary

user experience	usability	outlet connection
property	@IBOutlet	UIPickerView

### Materials

- **UnitConverter Lesson 9** Xcode project

### Opening

How can we get the temperature picker to start with a reasonable temperature selected?

## Agenda

- Discuss the user experience when first starting the app. Notice the default starting temperature in the picker view, and consider how it affects the user experience.
- Discuss where one might implement the behavior of specifying a default starting temperature.
- Using Interface Builder and the Assistant Editor (⌘⇧⌘), add the picker view as an `@IBOutlet` property within the `ViewController` class.

```
@IBOutlet weak var celsiusPicker: UIPickerView!
```

- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIPickerView` class reference and its methods `numberOfRowsInComponent:` and `selectRow:inComponent:animated:`.
- Set the default selected temperature in `viewDidLoad`.

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    let defaultPickerRow =  
        celsiusPicker.numberOfRowsInComponent(0) / 2  
    celsiusPicker.selectRow(defaultPickerRow, inComponent: 0,  
        animated: false)  
}
```

- Discuss the benefits of dividing the number of picker rows by 2, compared with using an explicit number (such as 100).
- Run the app (⌘R) and notice that, while the selected Celsius temperature has changed, the converted temperature label has not updated.
- Discuss the drawbacks of naively setting the label's contents to **32°F** explicitly in Interface Builder; and the drawbacks of explicitly updating the label in `viewDidLoad`, resulting in duplicate code already found in `pickerView:didSelectRow:inComponent:`.
- Implement the initial converted temperature in `viewDidLoad` by calling `pickerView:didSelectRow:inComponent:`.

```
pickerView(celsiusPicker, didSelectRow: defaultPickerRow,  
    inComponent: 0)
```

- Run the app (⌘R), observe the default selected temperature in the picker, and observe the converted temperature label.

## Closing

What happens when we select a temperature, background the app, and foreground the app?  
What happens when we force quit the app and restart it?

## Modifications And Extensions

- Add an "info" button to the view, and use a `UIAlertView` to allow the user to select the default starting temperature, saving the default value using `NSUserDefaults`.

## Resources

UIPickerView Class Reference [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIPickerView\\_Class/](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIPickerView_Class/)

UIPickerViewDelegate Protocol Reference [https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIPickerViewDelegate\\_Protocol/](https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIPickerViewDelegate_Protocol/)

The Swift Programming Language: Protocols [https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/Protocols.html](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/Protocols.html)

UIKit User Interface Catalog: Picker Views <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/UIKitUICatalog/UIPickerView.html>

Xcode Overview: Connecting User Interface Objects to Code [https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode\\_Overview/edit\\_user\\_interface.html#//apple\\_ref/doc/uid/TP40010215-CH6-SW3](https://developer.apple.com/library/ios/documentation/ToolsLanguages/Conceptual/Xcode_Overview/edit_user_interface.html#//apple_ref/doc/uid/TP40010215-CH6-SW3)

Interface Builder Connections Help: Creating an Outlet Connection [https://developer.apple.com/library/ios/recipes/xcode\\_help-IB\\_connections/chapters/CreatingOutlet.html](https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingOutlet.html)

Start Developing iOS Apps Today: Finding Information <https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/FindingInformation.html>