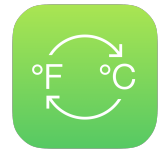


UnitConverter

Lesson 5



Description

Convert and display the selected temperature.

Learning Outcomes

- Apply Interface Builder connections to establish relationships between the view and controller.
- Implement operations with Swift numeric datatypes.
- Practice converting numeric types to accommodate desired operations.
- Apply string interpolation.
- Practice updating interface components with controller code.



Vocabulary

outlet	@IBOutlet	outlet connection
UILabel	type conversion	string interpolation

Materials

- **UnitConverter Lesson 5** Xcode project

Opening

How should we convert the temperature and display the temperature in Fahrenheit?

Agenda

- Discuss the requirement of converting and displaying the temperature when selected in the picker view.
- Using Interface Builder and the Assistant Editor (⌘⇧↔), create an outlet connection for the label as a controller property.

```
@IBOutlet weak var temperatureLabel: UILabel!
```

- Update the `ViewController pickerView:didSelectRow:inComponent:` method.

```
func pickerView(pickerView: UIPickerView, didSelectRow row: Int,
               inComponent component: Int) {
    let degreesCelsius = Float(temperatureValues[row])
    let degreesFahrenheit = 1.8 * degreesCelsius + 32.0
    temperatureLabel.text = "\(Int(degreesFahrenheit))°F"
}
```

- Discuss the need to convert the value retrieved from the `temperatureValues` array and the need to convert the converted temperature to an `Int` before updating the `UILabel text` property.
- Run the app (⌘R), select a temperature with the picker, and observe the converted temperature.

Closing

Although the basic functionality is complete, what problems do you see in our controller code? Is there a component that is missing from our project?

Modifications and Extensions

- Analyze the expressiveness of the code in `pickerView:didSelectRow:inComponent:` and extract the code into a well-named method.
- Refactor the temperature conversion and setting of the `UILabel text` property into a single statement, and critique its readability compared to the original code.

Resources

Interface Builder Connections Help: Creating an Outlet Connection https://developer.apple.com/library/ios/recipes/xcode_help-IB_connections/chapters/CreatingOutlet.html

Xcode Overview: Connect 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

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

UIPickerView Class Reference 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/

UILabel Class Reference http://developer.apple.com/library/ios/documentation/uikit/reference/UILabel_Class/Reference/UILabel.html

The Swift Programming Language: The Basics https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html

Swift Standard Library Reference: Numeric Types <https://developer.apple.com/library/ios/documentation/General/Reference/SwiftStandardLibraryReference/NumericTypes.html>