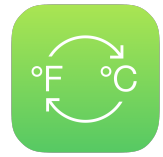


# UnitConverter

## Lesson 3

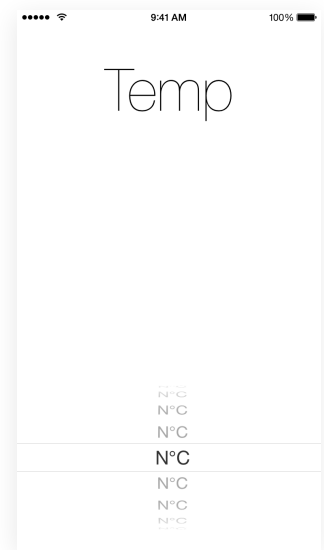


### Description

Adopt the UIPickerViewDelegate protocol, and implement methods to display placeholder data.

### Learning Outcomes

- Describe delegation and recognize the use of the delegate pattern in iOS apps.
- Practice using Interface Builder to set an interface component's delegate.
- Relate the use of protocols to specify a delegate's datatype.
- Apply a protocol declaration and implement methods that adhere to a delegate protocol.
- Practice creating custom breakpoints to provide insight about the state of a running program.



### Vocabulary

connection well	delegate	protocol
UIPickerViewDelegate	protocol adoption	unicode
breakpoint		

### Materials

- UnitConverter Lesson 3 Xcode project
- Delegates and Delegation presentation

## Opening

Why is the picker view displaying a bunch of question marks?

## Agenda

- Run the app (⌘R), and observe how the picker view displays the ? character.
- Explain how, without a delegate to determine what to display, the picker view renders a ? by default.
- Using Interface Builder, set the main View Controller as the picker view delegate by Control-clicking the picker view, and dragging a connection from the `delegate` connection well to the View Controller in the Document Outline (⌘⇧O).
- Present the concept of delegates and delegation.
- Discuss how delegation and protocols are often found together, and how an object delegates to objects that adopt a particular protocol, regardless of the actual type of the delegate object.
- Add the `UIPickerViewDelegate` protocol declaration to the controller class.

```
class ViewController: UIViewController, UIPickerViewDataSource,
    UIPickerViewDelegate {
```

- Using the Xcode Documentation and API Reference (⇧⌘0), explore the `UIPickerViewDelegate Protocol Reference` and the methods `pickerView:titleForRow:forComponent:` and `pickerView:didSelectRow:inComponent:`.
- In the `ViewController` class, implement `pickerView:titleForRow:forComponent:`.

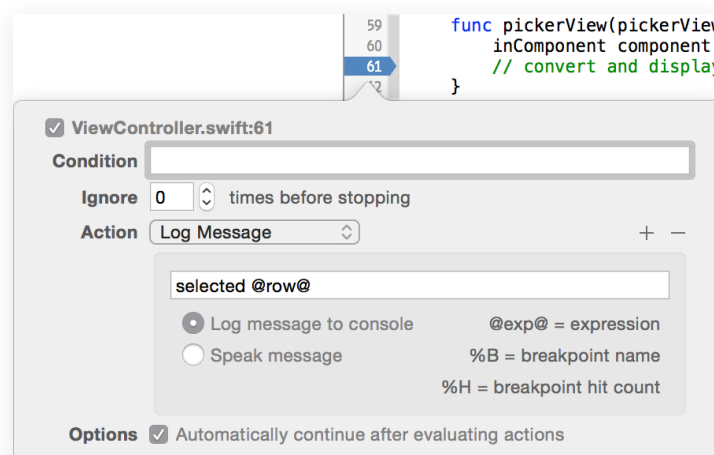
```
func pickerView(pickerView: UIPickerView, titleForRow row: Int,
    forComponent component: Int) -> String! {
    return "N°C"
}
```

- Discuss the the unicode ° character.
- In the `ViewController` class, implement `pickerView:didSelectRow:inComponent:`.

```
func pickerView(pickerView: UIPickerView, didSelectRow row: Int,
    inComponent component: Int) {
    // convert and display temperature
}
```

- Explain the relationship between the two protocol methods and the picker view.

- Explain the arguments the picker view will pass to the `pickerView:titleForRow:forComponent:` and `pickerView:didSelectRow:inComponent:` methods.
- Add a custom breakpoint to `pickerView:didSelectRow:inComponent:` that generates a Log message containing selected: `@row@`.



- Run the app (⌘R), observe the values displayed in the picker view, flick the picker to select a row, and observe the console (⇧⌘C) message when the row is selected.
- Discuss the values of `row` that appear in the console, and how the rows in the picker view are zero-indexed, similar to arrays.

## Closing

What do you think we need to do to get the picker to display rows of temperature values?

## Modifications And Extensions

- Explore the `UIPickerViewDelegate` method `pickerView:viewForRow:forComponent:reusingView:` and customize the appearance of the rows in the picker view.

## Resources

Cocoa Core Competencies: Delegation <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Delegation.html>

Start Developing iOS Apps Today: Delegation [https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/DesignPatterns.html#//apple\\_ref/doc/uid/TP40011343-CH5-SW4](https://developer.apple.com/library/ios/referencelibrary/GettingStarted/RoadMapiOS/DesignPatterns.html#//apple_ref/doc/uid/TP40011343-CH5-SW4)

Core Cocoa Competencies: Protocol <https://developer.apple.com/library/ios/documentation/General/Conceptual/DevPedia-CocoaCore/Protocol.html>

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)

Interface Builder Object and Media Help: Setting an Object's Delegate [https://developer.apple.com/library/mac/recipes/xcode\\_help-IB\\_objects\\_media/Chapters/set\\_object\\_delegate.html](https://developer.apple.com/library/mac/recipes/xcode_help-IB_objects_media/Chapters/set_object_delegate.html)

Concepts in Objective-C Programming: Delegates and Data Sources <http://developer.apple.com/library/ios/documentation/general/conceptual/CocoaEncyclopedia/DelegatesandDataSources/DelegatesandDataSources.html>

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/)

Source Editor Help: Adding, Disabling and Deleting Breakpoints [https://developer.apple.com/library/ios/recipes/xcode\\_help-source\\_editor/chapters/Creating,Disabling,andDeletingBreakpoints.html](https://developer.apple.com/library/ios/recipes/xcode_help-source_editor/chapters/Creating,Disabling,andDeletingBreakpoints.html)

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